Remodelling a Credit Management System
using an Information Systems Approach

Imran Sarwar
Accounting & Information Systems

2003/2004

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student) _____

Summary

To manage a firm a good control of credit is needed. Poor usability and interface design can lead to user errors which can cost a firm. This project was the result of research into credit management and control. It is a system which benefits from modern day GUI and is designed to help the credit manager.

The problem I will solve is to see how issues such as usability, poor interface design and other human factors impact the system.

This project was designed as a learning aid to the credit managers I worked with to show them the advantages of a better usability**.**

Acknowledgements

Without the help of some people this project would have suffered. The help from Anne Phillips (Credit Manager) allowed me to getter a better view into the credit management world. I would also like to thank all those at Hypercom who letting me test their system and providing me with guidance.

Other people Id like to acknowledge for their help;
Natasha Shaklevich for her support and suggestions
Antony for the cups of tea
All credit controllers and managers involved in the testing.
All the students for helping during the testing during development.
And anybody else who I cant remember at this time.

CONTENTS

Chapter 1	**Introduction**
1.0	Introduction to project
1.1	Objectives of project
1.2	Minimum deliverables
1.3	Introduction to Credit management
1.4	Stakeholders and who will evaluate
1.5	Scope of the project
Chapter 2	**Analysis**
2.1	Research
2.1.1	Research methods
2.1.2	The Sector and Company A
2.2	Feasibility study
2.2.1	Problem definition
2.2.2	Success factors
2.2.3	Requirements
2.2.4	Current solution and alternative solutions
2.2.5	Preferred solution
2.2.6	Technical feasibility
2.2.7	Social and Legal feasibility
2.2.8	Organisational
2.2.9	Economic
2.3	Constraints
2.4	Problems
2.5	Current systems and strategy
Chapter 3	**Methodology**
3.1	Introduction to Methodology
3.2	What will be included in the project?
3.3	Project Schedule
Chapter 4	**Requirements Analysis**
4.1	Identifying Requirements
4.2	Functional Requirements of system
4.3	Non functional Requirements
Chapter 5	**Human Issues and Human Computer Interaction**
5.1	Human Issues
5.2	Usage Costs
5.3	Interface
5.4	Usability
Chapter 6	**Design and implementation**
6.1	An Introduction to Design and Implementation
6.2	An introduction to the Database
6.3	Normalisation
6.3.1	First Normal Form
6.3.2	Second Normal Form
6.3.4	Third normal Form
6.4	Database Schema
6.4.1	Integrity
6.5	Choice for Prototype
6.6	Use of Access Wizards
6.7	Tables
6.8	Queries
6.9	Security

6.10 Form and Subform Design
6.11 Reports
Chapter 7 **Testing**
7.1 Introduction to Testing
7.2 Database testing
7.3 Data entry/Form Testing/ Statements and Letters
7.4 Task based Testing
7.5 System Testing
Chapter 8 **Evaluation**
8.1 What has been achieved and Evaluation of objectives
8.2 Evaluation of prototype
8.3 Further development
8.4 Time
Appendix A **Reflection**
Appendix B **Gantt Chart**
Appendix C **Interview with credit manager**
Appendix D **Screen Designs**
Appendix E **Dataflow diagram of scenario**
Appendix F **Acceptance testing form**
Appendix G **User Manual**
Appendix H **Evaluation feedback form**
Appendix I **Sun Screenshots Vs. New prototype**
Appendix J **VBA Code and SQL queries**
Appendix K **Example out put letter and statement.**

Chapter 1 Introduction

1.0 Introduction to project

As businesses become more competitive the need to maintain a steady cash flow becomes vital. This is what is known as credit management. Following my time working in credit management/control I came across many disgruntled employees who would criticise their systems. With the help of these disgruntled credit managers and credit controllers it became possible to begin remodelling the credit management system using approaches learnt in information systems and accounting. Credit managers felt that their current systems were at the end of their lifecycles and had started to become a cost to their companies. This project is designed to aid them into understanding the requirements and just how a solution can help their work; it will therefore be evaluated by manager and controllers. The aim of the project is to analyse the current problems and how one current solution deals with the problem. Combined with research into the area this will result in a prototype which can be presented to credit managers for evaluation.

It was not possible to use a named company for this project due to data protection and competition issues. Therefore the scenario is based on the company's and sectors the credit managers and controllers work for. The industry sector is the sale and leasing of credit card terminals. Throughout this project the company shall be known as Company A.

1.1 Objectives of project

- Research into credit management and how firms deal with the issue of credit management and control.
- Basic feasibility study to explore the problem.
- Explore current solution used and evaluate how current solutions meet requirements.
- Carry out interview with credit managers/credit controllers to identify core requirements and possible usability issues.
- Research and explore different modelling tools to model problems with.
- Design prototype screen designs and test with potential users.
- Test prototype solution with potential users.
- To explore how human issues may affect the system.

- To evaluate the solution and review how it may improve business processes.
- Comment on my findings and suggest improvements for any future implementations.

1.2 Minimum deliverables

These are the deliverables that are needed to measure the success of the project.

- Feasibility study and Requirements analysis.
- Compare and select the best tool to model the current systems and use the modelling tools I learn to design the proposed system.
- Analyse how current systems help the strategy of the business and whether or not they hinder the business in any way.
- Propose a prototype model which will benefit from the design aspects I learn in human computing and include them into my model. This will take the form of a demonstration model which will be a basic implementation of the proposed system.
- Evaluation of the solution and the project to see if all the requirements have been met and to explain any changes which were needed.

1.3 Introduction to Credit management

Edwards (1997) confirmed that "Credit management is still regarded in many companies as solely the collecting of overdue debts. Companies deal in credit, where credit is given it must be collected. This is the main focus of this project. Background research comprises of my experience in credit control and interviews with credit managers as well as consulting literature. I consider this as more relevant research as it involves the first hand user problems and how they deal with real problems. I found only one book in the Leeds university library on Credit management and will therefore need to use journals and online resources for research. Considerations of new methods or changing practices in the credit management world will be investigated by viewing the Institute of Credit Management website at www.icm.org.uk.

Credit management is a technique of managing the company's flow of funds. Businesses would certainly come to halt if everything was to be paid by cash on collection. Credit departments can be split into giving credit and collecting credit. There are costs involved in managing credit. The value of the sale will decrease the

longer it remains unpaid affecting cash flows. This is why businesses need systems which are dedicated to credit control and provide tools to aid credit controlling. This is often overlooked with companies choosing to use applications such as Excel to form sales ledgers. The effects of poor credit management include inability to meet payments and direct erosion of profits available for use within the company. Poor credit management is also a sign of poor liquidity and signs of trouble within a company. All to which its staff are sensitive to and respond accordingly.

Users of a typical credit management system are the credit managers who use it for account ageing analysis and credit controllers who will use the system to chase funds and talking to customers.

1.4 Stakeholders and who will evaluate

The target stakeholders are the credit managers who I am doing this study to aid. The outcome can influence their choice whether to invest in a new system or stick with their current one.

Testing will be done to test the system and evaluate the design and functionality. This is to ensure that usability, functional and non functional requirements have been satisfied. Users will range in experience from typical users of the current system to someone new to credit control.

1.5 Scope of the project

A full scale implementation is outside the scope of the system. The prototype is intended to show the users how design issues and functionality can be enhanced to aid working practices. The system will be demonstrated managers and any implementation would have to use a simulated database due to data protection issues.

Chapter 2 Analysis

2.1 Research

To fully understand a problem research is needed to ascertain user requirements and system requirements. These all help into identifying expectations from a final system.

2.1.1 Research methods

1) Overview of literature relating to credit management

2) Testing current systems used in the industry: Having worked in credit control for a period of time I have come into contact with many systems. These systems I will use to form data flow diagrams and form the basis of the prototype.

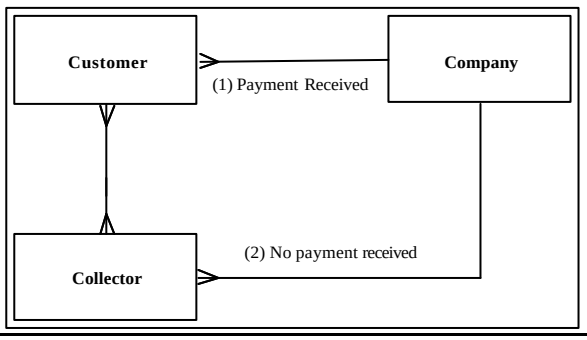
3) Interviews: Primary research is very important in requirements capture. Therefore I have three credit managers and two credit controllers who I will use as the foundation of my research. A transcript of one of the interviews can be found in the appendix C.

2.1.2 The Sector and Company A

The sector I will be using deals with the sale/lease of credit card terminals. For competition reasons their name cannot be divulged. They will therefore be referred to as Company A. It has an accounts team which consists of one permanent credit manager, one temporary credit manager and two credit controllers. There is a separate role for the data input as well as a cancellations department. These roles will be generalised into the credit controller's role for the purpose of this project. Company A recently signed off a debt clearance project to clear up outstanding debts which have accrued over 3 years, which I was involved in. A flaw in their system resulted in them not picking up this error and consequently found them being owed in the region of £4 million. The company issues one invoice per year which covers four quarters lease. Customers need to give 30 days notice for cancellation. Where there have been account closures and cancellations of direct debits there has been a problem in getting customers to pay what is due. All payments are recorded on spreadsheets currently and they feel no need to integrate this to the credit management system.

ER diagram of Customer, Company and Collector

Figure 1



Company A uses Sun systems tailored by InHowes and Collect designed for Sun. Reporting is done using MS Excel XP through Vision. This pulls data from the database onto a report.

Another company I will use for some research is Company B, a credit collections company. Their system is an outdated system linked to a dialler system. The note taking feature is very primitive allowing only 3 lines of input. Their collection techniques are what will be used as the basis of a new system. A generic package solution such as Sage line 50 or 100 is an alternative that small companies use but these firms are too big for such generic packages.

Merise Flow Diagram of Credit Control Department.

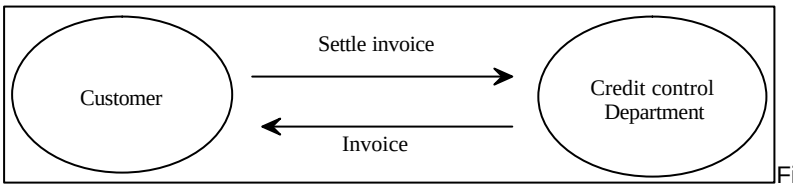


Figure 2

2.2 Feasibility study

2.2.1 Problem definition

The problem is one of a legacy system with lots of islands of automation and no integration. The integration is not in the scope of the project but the legacy system dealing with the credit management is. Company A which I am basing my research on uses Sun systems by InHowes as their main system with Collect for Sun as an add-on. This add-on was designed to meet the changing needs of the company when Sun systems was seen as failing. A full scale redesign and implementation is unfeasible now because of the costs and time limitations. A project of this scale would take many years. The aim is to remodel the solution and propose a prototype which is more usable and more relevant to today's problems such as interface design.

2.2.2 Success factors

A success factor would be a tangible savings in the collections process. A return on investment could be calculated by calculating the difference in collection amounts. Another success factor is to ensure that the relationship between the data is modelled correctly. Other factors can be found in the interview in appendix c. Success factors are identified as:

- A usable system
- Easy to use and with a user friendly interface
- Information wealth on screen.

2.2.3 Requirements

The system in use by company A has been adapted over time to meet changing requirements. Upgrading and maintenance tasks have led to a lot of unusable data. Changes in working practices and a high staff turnover result in new practices being put in and the old being forgotten.

E.g. Invoices now use the prefix Month+Number so when running reports looking for unpaid invoices anything with any other prefix is not picked up. Problems such as this occur when legacy systems are continually updated to increase their life cycle.

2.2.4 Current solution and alternative solutions

The current solution is Sun systems by InHowes. This is well into its decline phase and can barely be described as a support system in the state it's in.

1) An alternative would be to look at some of the systems in use and design an alternative based upon current needs of the credit controller and the credit manager. This solution would review many of the current systems, analyse them and propose new data models of the current activities. The prototype would hopefully be presented to credit managers for evaluation so that they could see another way of achieving their goals.

2) No system needs to be developed; existing applications such as spreadsheets could be used as a sales ledger and invoice information. This could also be stored as a paper based solution. This is a possible solution but would provide no competitive advantage and is a poor information systems strategy.

Spreadsheet Design taken from Bass (1991)

A/C number	Name	Risk Cat	Credit limit	Terms	Total	Balance	Current	1-30 days	31-60 days	Over 90	Notes

Figure 3

3) Invest in another generic package such as Sage line 50.

2.2.5 Preferred solution

Solution 1 is preferred. Although no current plans to remove the current system are in place. The development of the prototype may persuade them into trying to secure investment for an improved system.

2.2.6 Technical feasibility

Any skills which are needed to develop the prototype will need to be learnt in my own time.

From my university studies I have an understanding of SQL and simple programming. Any further knowledge will have to be learnt. The proposed prototype is for testing and evaluation purposes as a graphical representation of the prototype. It is important an over complex system is not designed as this will alienate the users. It is more important that it is designed properly so that it can be used effectively and efficiently. Current infrastructure and user ability is sufficient for testing purposes.

2.2.7 Social and Legal feasibility

Acceptance is important in any project. Sauer (1993) lists acceptance as one of the most important reasons for project failure. For user acceptance it is important for clarity over the objectives of the project. Therefore managers need to be made aware of the goals of the system and how this will contribute to its success. Credit controllers are important for investigating issues such as usability, HCI and analysing GUI.

The current system is also computerised so a certain level of skill is presumed. As the prototype will not be implemented into any company further social needs can be ignored.

Any data which is used for test purposes cannot be taken from any of the companies databases. Therefore any screen shots of current systems black out any strategic information or customer details. All data exchanges must comply with the data protection act.

2.2.8 Organisational

It is important to secure management opinions and approval for the prototype. So research will involve interviews and meetings with managers to get a better requirements capture. Issues such as security are outside the scope of the prototype. The focus is more on getting the data modelling and ensuring integrity of the database design.

2.2.9 Economic

The only proposition is a prototype which has no cost to the manager's.
If after viewing managers decide to undertake a similar project then its real value as a prototype has been served. The underlying aim is to get the managers to change the practices of the company they work for. This will improve their application portfolio and get them to view IS as a way to gain competitive advantage not a

means to an end. User satisfaction is hard to quantify but the cost of poor design according to Bennet, Mcrobb and Farmer (1999) could be increased frustration, increased error rate which could lead to loss of confidence and absenteeism or higher staff turnover. Financial example in Chapter 5; Human Issues.

2.3 Constraints

It is ambitious to think that a project done by a student can be implemented full scale into a multi million pound company. Therefore, the scope lies within showing managers and controllers a new way of working and hopefully using the ideas to formulate their own solutions. Time is also limited as is the resources I can devote to the project due to commitments to other modules.

2.4 Problems

Problems include getting the user requirements correct and then modelling them. Poor requirements capture could lead to system which does not meet its core requirements which will lead to a solution which fails usability.

2.5 Current systems and strategy

Systems in company A are there to help it with day to day activities. However, in terms of cash flow they should be seeking a competitive advantage. Conducting a current systems audit most of the applications can be seen to be in the support section rather than key operational where most accounting packages should be. As demonstrated in figure 4.

Following an interview with one of the credit managers it was clear that their main system Sun was well into its decline and upgrades and maintenance had only extended its life.

Current Application portfolio for company A based on Mcfarlan (1984)

Strategic	High Potential
Pos terminal network	New credit card terminals
Key operational	**Support**
Stock control system	Accounts payable(Sun)
	Accounts Receivable (Sun and Collect)
	Spreadsheet applications
	Word Applications

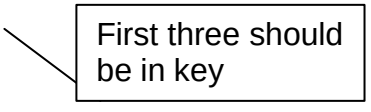


Figure 4

Company A have got the option to replace Sun with another generic off the shelf package such as Sage line 50 or similar. These are low cost solutions and can be tailored to their needs, but provide no competitive advantage in collection.

Credit managers must see that the current system does hinder their work and it is now costing them in terms of competitive advantage and poor cash flow, time, training as well as maintenance costs. Good Management information systems need good data processing systems. A company such as this needs to have its IS strategy not stuck in the early data processing era.

A good credit management system can help them achieve their business goals by aligning strategic goals with there IS/IT needs. The main aim is competitive advantage and good cash flow is one example of this advantage. Ives and Learmonth (1984) and Galliers (1987) added that low quality of IS/IT and no vision = “safe” support environment. Poor IS/IT with a Vision it is defined as “wasted” high potential environment. Currently the systems reflect poor investment. Credit managers viewing this project should be able to see how changes to the interface can provide them a lot higher payback then they think.

Chapter 3 Methodology

3.1 Introduction to Methodology

One of the best questions in developing a system is what methodology to use. I feel Priestly (2000) echoes my reasons when he says that “Systems are rarely developed according to the strict linear process specified by traditional design methodologies”.

Following a review of methodologies used in project management I have decided to use a combination of a few. The main methodology is the waterfall model along with the prototyping approach. The biggest problem with the waterfall model is that problems with the project are not often found until the testing stage which I feel is too late and by that time a lot of resources have been committed already. The waterfall model also suffers from requirements being committed very early on in the project. This does not allow any requirements evolution or flexibility. This I feel might hinder the project because management practices are always changing and hence the system should not be out of date before it is delivered.

So I will use these aspects from different methodologies; Problem Definition, Feasibility Study, Requirements Analysis and Design from the Waterfall model. I will use this model in its iterative form as it complements the prototyping approach. These are then reviewed and approved before the system enters its final stages. This promotes any ownership aspects of the system and regular reviews can ensure that the project is still on target. More information on these can be found in McConnell (1996).

Requirements analysis will be done using UML and other object orientated tools. The object orientated approach is the more favoured way currently in the industry to model requirements. Also as a joint honours student it is better to experiment with more than one methodology in order to make the project easier to comprehend for different stakeholders.

Table comparing different methodologies considered (Fig. 5)

Methodology	Main advantages	Main disadvantages
Waterfall model	Each stage is planned and follows other stages. Used in major defence projects. It is tried and tested and can be adapted.	Problems are often not found until late in the project. Very expensive if things go wrong.
Rapid Application Development	Iterative approach and produces solution quicker than Waterfall model	Still a build quick and fix, fix, fix approach.
Rational Unified Process	Modern type of methodology, iterative, using UML.	Difficult for first time users and expensive to find people with knowledge of it.

3.2 What will be included in the project?

These areas will be covered in the project. 1) Definition of Problem 2) Background research 3) Feasibility study 4) Requirements analysis 5) Design and implementation 6) Prototype 7) Testing 8) Evaluation.

Tools to help me achieve this will be; 1) Entity Relation Diagrams 2) UML case and Class diagrams 3) Data flow diagrams 4) Process flow charts amongst others.

One of the extensions to the minimum requirements is an implementation which will consist of a basic prototype which will demonstrate the invoice analysis benefits of the proposed system.

Kendall and Kendall (1999) suggest that "Prototyping of Information systems is a worthwhile technique for gathering specific user requirements". They go on to define four types of prototypes, which are the

1) Patched up prototype
2) Non operational prototype
3) Selected features
4) First of a series

Out of these the selected features prototype is the one to be used.

3.3 Project Schedule

The project schedule is in the form of a Gantt chart (appendix b). This highlights the major stages of the project and when I plan to complete them. Any deviations will be discussed in the evaluation. The Major stages are when a minimum requirement is met. However, the stages that contribute towards this milestone are also addressed.

During the Christmas period to gain more experience on a credit management system I will be working as a credit controller using Sun systems and Collect at Company A.I will use this opportunity to document some advantages and drawbacks of their system. During this time I will also take the opportunity to talk to primary users of that system.

Chapter 4 Requirements Analysis

4.1 Identifying Requirements

Bennet, Mcrobb & Farmer (1999) state that “Users requests change over time” and that analysis is what happens in the current system and what is required of the new system. Therefore Requirements capture is important so that the stakeholders can assess the requirements understanding before signing off the implementation. Changing requirements usually signify high costs in systems development. I adopted a methodology which would still allow me to make some changes. Flynn (1992) Identifies requirements determination to consist of these areas; problem definition, feasibility study, requirements acquisitions and a requirements analysis. These areas all need to be addressed. The requirements acquisition is done through an interview (Appendix C).

One of the minimum requirements is to explore different tools for modelling which will be used in this section. In the requirements section is it important to view the problem as it is on the outside. First of all and ER model needs to be done of the scenario. ER modelling classifies everything in to attributes, entitities and relationships.

Chen ER model of Scenario

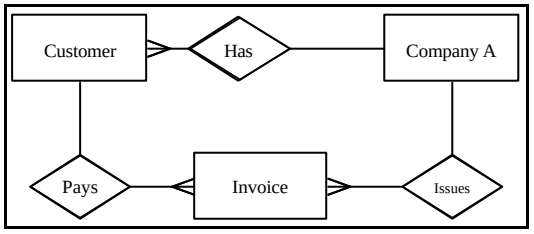


Figure 6

Analysis of current systems showed that there was a common flow of activity within the systems. Data capture points need to be identified for interface requirements purposes. Most current systems have common input needs which are customer details, product details and invoice details. The payment status separates them from being paid or unpaid. For credit controllers the unpaid invoices is where the wealth of information is needed.

Dataflow through a CM system

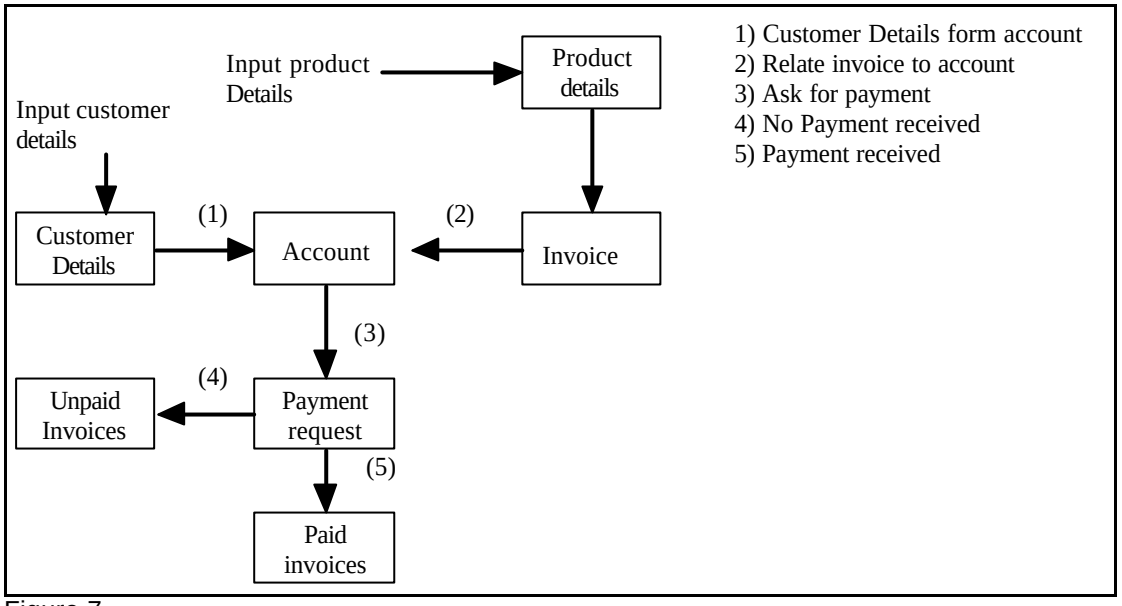
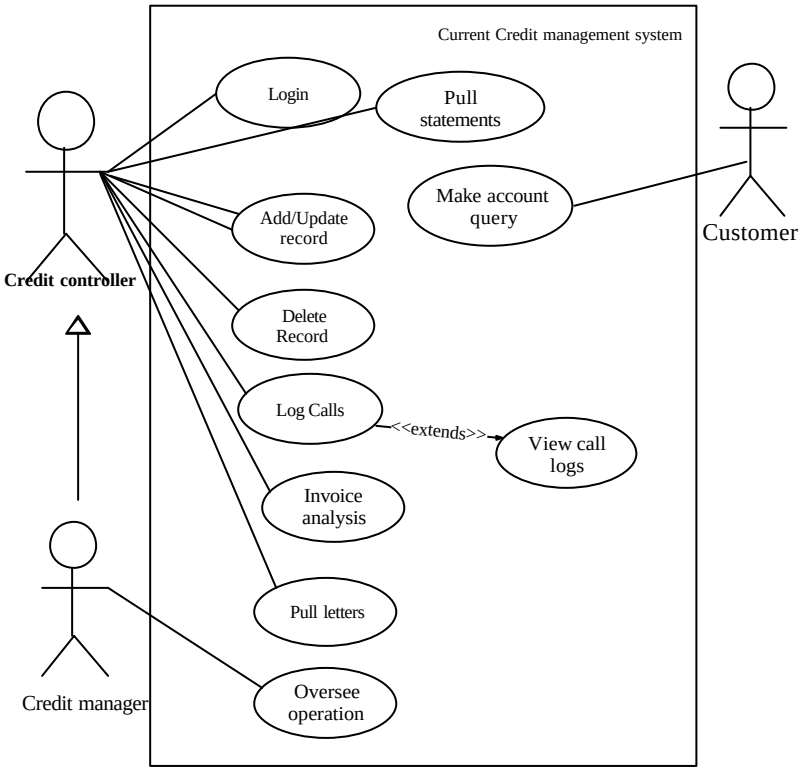


Figure 7

To model the requirements UML was used for its graphical abilities to illustrate requirements. This was preferred to the traditional requirements analysis document which is lengthy. This is an easy to understand diagram which managers can use to sign off requirements specifications.

UML case diagram of Proposed system.



In Scope of prototype:
- Statements
-Add/Delete/Update
- Log calls/ ViewLog
- Invoice analysis

Not in prototypes scope
- Security

Figure 8

These use cases can then be used in determining what the scope of the proposed system is. The use cases identified above would be needed in any implementation of a credit management system. They identify the aspects needed for a credit collector to do their job. Hence, if a system does not meet these requirements then it is a failed solution. These can be translated into functional requirements of the system The scenario then needed to be modelled in a simplified UML class diagram to show any specialisation/generalisation in entities.

UML Class diagram to show scenario.

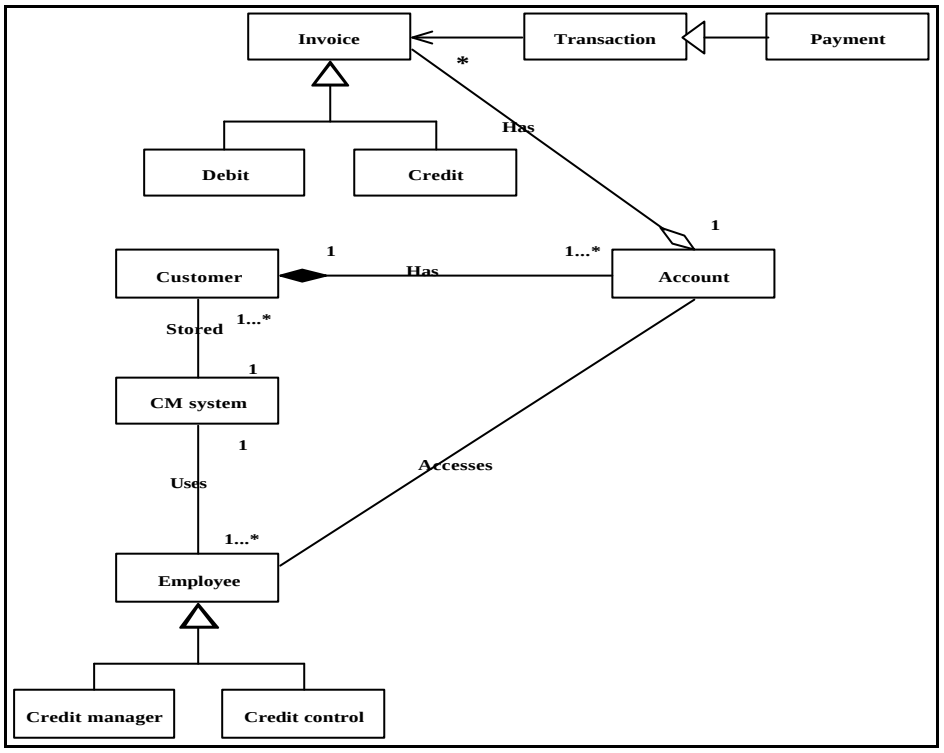


Figure 9

If the data is valid then the data needs to be processed to form meaningful information. The output will be on screen or paper output. The following diagram illustrates the inputs and outputs required by any credit management solution.

Input/Output diagram

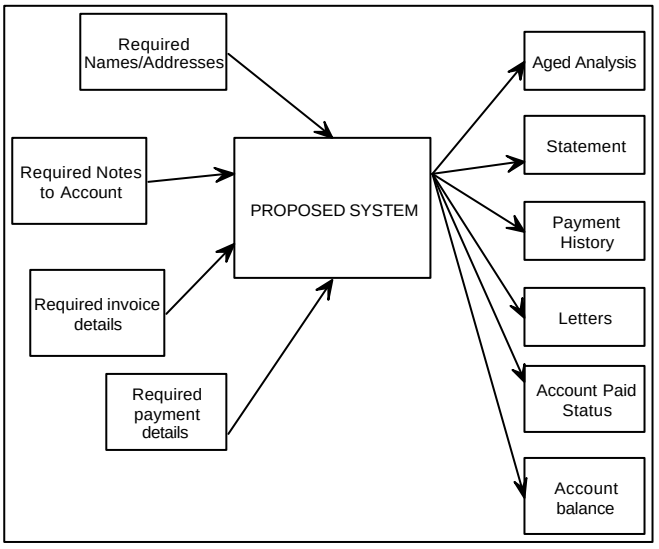


Figure 10

The system needs customer details and invoice details so that it can relate the two together to form an account for the customer. This flowchart demonstrates the flow of activities in the system and outside it.

UML activity diagram with swim lanes

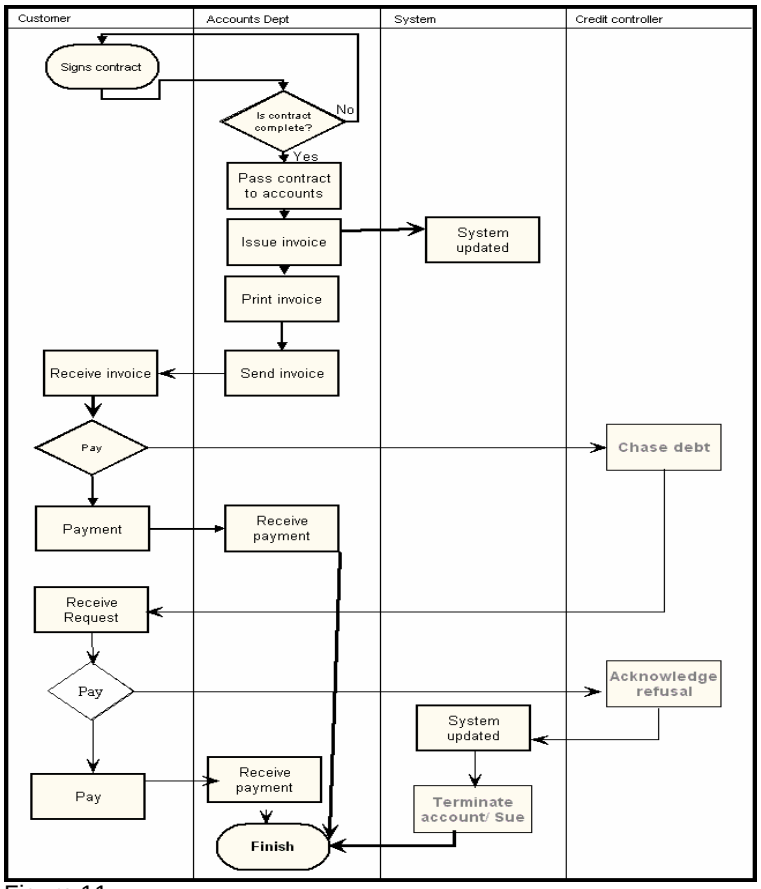


Figure 11

A flow chart can also be done of processes such as adding customers to the system. This is documented below. This would show what requirements are captured and when. Exceptions can be shown in these process charts. Bailey (1989) finds that if functionality is not up to scratch then the human interface design will not have an impact. Therefore processes such as adding customer and invoices need to examined to get a clear requirements capture. A dataflow diagram of the whole scenario can be found in Appendix E.

Adding a customer to the system

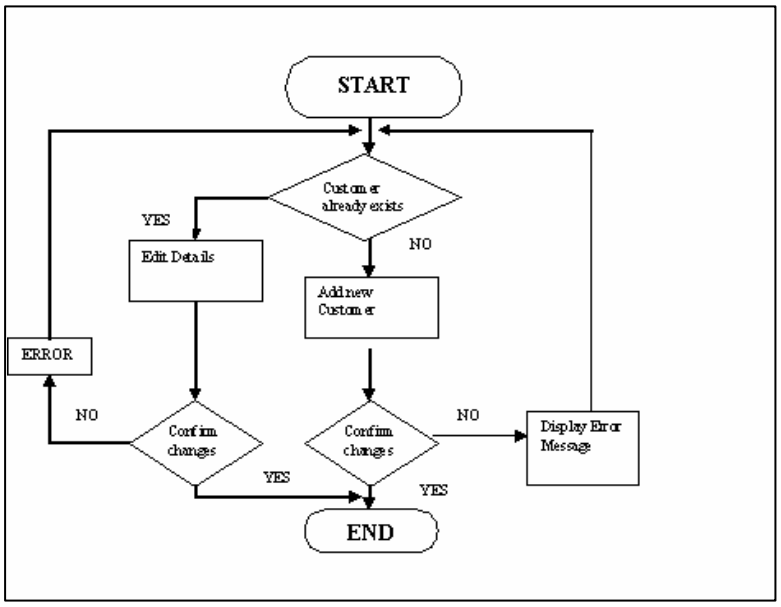


Figure 12

Customer is added if they don't already exist already. It is possible for many customers to have many accounts which can lead to collections problems. Where possible it should be tried to be kept to a one to one relationship with all invoices on one account. From the analysis these are the requirements which I formulated

4.2 Functional Requirements of system.

1) Tracking features for accounts which have been unpaid
2) Ageing tools for accounts
3) Ability to add, delete and update accounts.
4) Search facility to find accounts and notes.
5) To be able to have all relevant information on one screen when chasing accounts rather than searching through lots of screens.
6) Analysis of account more than 90 days old

4.3 Non functional Requirements

1) Usability
2) Graphical user interface
3) User friendly and easy to use
4) Minimum training required to use system

Chapter 5 Human Issues and Human Computer Interaction

5.1 Human Issues

HCI is the way we interact with computers. It is the backbone of most information systems. When designing the system it is important to realise the benefits a well designed interface can yield. More systems are now designed with a people centred approach. Factors such as usability of a system are now a major part of the systems development. The system is designed for a credit controller to use. Therefore it needs to be designed in a way to optimise their current experience and skill. This can be done through usability engineering to maximise on characteristics such as learnability and generalisability. Interface problems are often the biggest criticism of systems. The evaluators of the project already use a current system therefore a certain level of skill can be assumed. Most of the users will tend to be primary users.

Schneiderman (1992) points out in his literature that operator training time is expensive so the ease of learning is important. Eason (1984) suggests major indicator to be whether system is being used or not. However, most users will be primary users who will have to use the system as part of their work so this is an unreasonable assumption by Eason.

5.2 Usage Costs

Currently it takes the credit controller on average 60 seconds to load all the forms he needs to deal with a customer query. Each new form closes the previous and all the data has to be input again.

A simple calculation:

- Assuming 60 seconds to load forms per call with 15 calls per hour. Each of 4 minutes long (1 minute loading time)
- 8 hour working day and wage rate of £7.50 an hour.

A = 15 x 3 minute calls = 45 minutes per hour spent on calls

B = 15 x 1 minute loading per call = 15 minutes per hour wasted on loading

C = 8 hrs per working day X B = 2 hours wasted per day on interface

D = C X £7.50 = £15.00

Total cost of poor interface per day per controller is £15.00.An improvement of only 15 seconds per call could lead to a saving of £2.50 a Day per credit controller.

5.3 Interface

In the design chapter you saw the screen designs and how they were chosen. This section is more relevant to putting information where the user would expect to find it.

E.g. A user may expect to find customer phone number in customers screen but not invoice details (Fig. 19)
One of the deliverables is demonstration model. Its true success will be measured in its interface design and how well it can be used. To gain user acceptance the system had to be designed to boost user confidence and not alienate them with over complex screens which were echoed in the interview in appendix C.

5.4 Usability

Lingaard (1994) defines usability as Learnability, Flexibility, Effectiveness and Attitude. A good graphical user interface will play to usability factors such as learnability. When choosing the right layout for the GUI a lot of research was done into usability and many screen mock-ups were done (App. D). The best design was then used in the prototype. It is important to have a common language during the requirements capture so that issues such as expected skill level can be ascertained. It is assumed that most of the users will have used Microsoft Windows before, so in order to use this generalisability factor it is important for forms and button to do the same as they would in the operating system. Consistency will also help in improving user understanding. Learnability is the most useful in this project as it cuts down learning time and minimises errors.

Lack of usability can lead to user dissatisfaction. A loss in user confidence can impact user acceptance ultimately affecting the projects success. The technology must therefore do the job, but creating the culture with in which it exists is its true value. Lingaard (1994) suggests that usability defects fall into categories such as screen navigation, screen design and layout, user control, consistency and feedback amongst other.

6.1 An Introduction to Design and Implementation

Rambaugh (1997) defines design as “How it will be constructed without actually building it”. Using the prototyping approach meant it was being iteratively built.

It was determined early on the project that the data I was going to use for the model needed to be simulated due to the nature of the data that was stored on the current systems. The database would therefore be a simulation and not break the Data Protection Act. This is one of the reasons why the database design is not a major issue in the evaluation unlike the interface.

The credit managers were able to supply me with the requirements of a system (see requirements section) from here I was able to design the database for the documented scenario. Entity Relationship models differ from company to company depending upon their implementation. The ER models included in this section are relating to the simulated database I designed. I feel the more important element of the project is the interface design than the database design. The aim of this project is to show the credit managers how a system can aid their work, and I feel in depth database considerations will be out of the scope of this project.

ER diagram for customer relation (Chen)

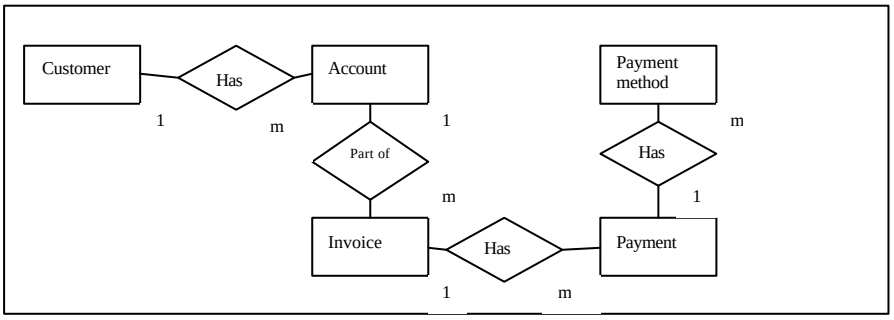


Figure 13

Add product dataflow (Gane and Sarson)

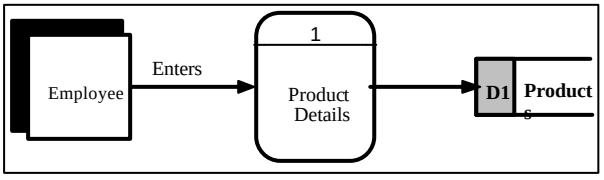


Figure 14

Adding a customer

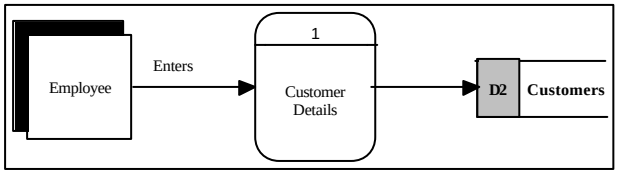


Figure 15

Entering an invoice

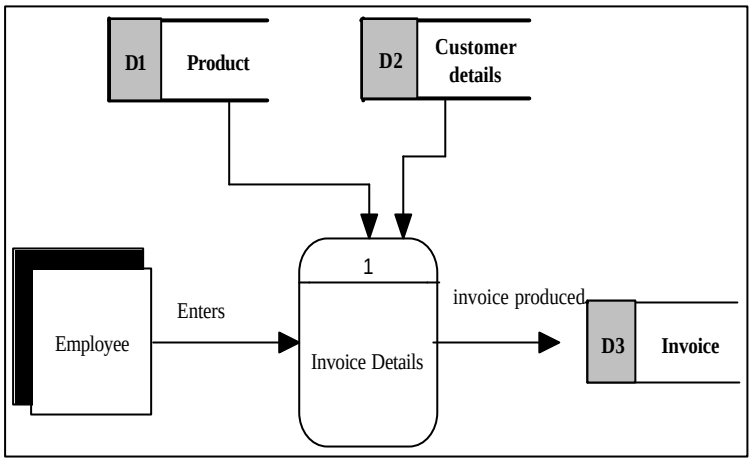


Figure16

6.2 An introduction to the Database

Elmasri and Navathe (2000) definition of a database is “A Database is a collection of related data”. To describe the inner workings and optimisation of databases was not within the scope of this project. A basic knowledge of database fundamentals is assumed. However, the credit management sector consists of many firms all which have very different databases. Therefore this chapter will deal with the simulated database I designed for the purpose of this project. A prior knowledge of databases is assumed.

The database allows the use of relations which a spreadsheet would have problems with. The database allows improved integrity and the ability to reduce data redundancy and inconsistency over spreadsheets. The relational model can be described as a model that provides a logical connection among tables by including data from one table in another table.

6.3 Normalisation

Normalisation which was first proposed by Codd (1972a) takes a relation schema and tests it to see if it complies with a certain normal form. For more information on these please see Elmasri and Navthe (2000).

6.3.1 First Normal Form

To meet first normal form the database must not have any repeating groups. This form is considered by many to be the formal definition of a relational model.

6.3.2 Second Normal Form

The basic concept behind this is the “concept of full functional dependency” as stated by Elmasri and Navathe (2000).

This is achieved by having separate tables for sets of values that apply to multiple records. These are then related with a foreign key.

6.3.4 Third normal Form

3NFis based on the idea of transitive dependency. This will mean any field that does not depend on a key (non key) should be eliminated. Also another field (non key) cannot functionally determine another non key field.

There are also many other forms which are theoretically desirable. Nonetheless, having a table for each field e.g. a table for cities and post codes etc may in turn degrade performance whilst trying to achieve the strongest normal form. BCNF and 5NF do improve database design but for the database I am going to design it is understood not to affect functionality much. Generally designers use $3^{NF\ only}$ to data that changes frequently

The tables used will at least meet 1NF; further normalisation will only occur if I feel the database design is suffering. The main point for testing and evaluation is the interface not the database design.

6.4 Database Schema

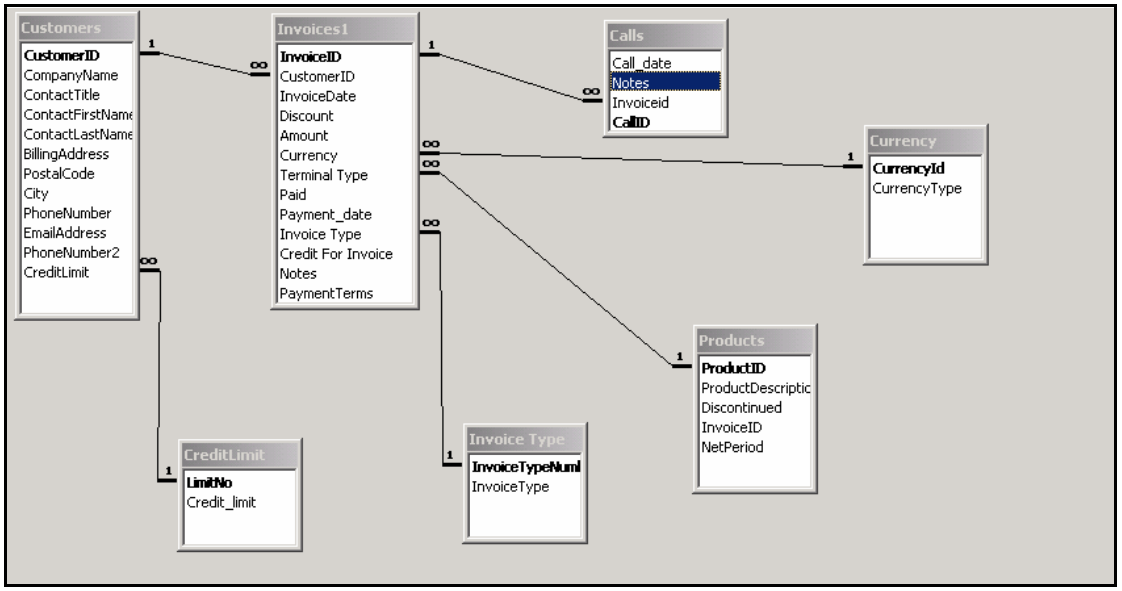


Figure 16.5

Note: A Call note is different from invoices1.notes. Call notes are notes made on phone calls whereas Invoices1.calls is special notes on invoices.

6.4.1 Integrity

Referential Integrity as well as Data integrity are very important in databases , especially when relationships are defined. In Access the use has the ability to force referential integrity. This will make sure cascading changes do not happen and data

field that are linked match. This approach was used to make sure that the tables met referential integrity rules.

6.5 Choice for Prototype

Originally my choice for the database was SQL server, it provided the functionality I needed along with the stability which is desired from a database. However, due to many of the test sites not having SQL Server I opted to carry out the prototype in Microsoft Access instead. One of Access biggest advantage was that as a demonstration model its clarity would aid the understanding of the credit managers. My first choice for the prototype was SQL server. Portability was an issue as demonstration could be at work or at home, so as a stand alone system Access had what I needed. With it also being a program that most credit managers had used before they were more confident in suggesting changes and improvements. Access is limited in what it can do. It is designed for the single user or a small business. The next step up would be SQL Server. Regardless of this, as a vessel to produce a prototype it is however very valuable. It is hoped that using a program that some of the credit managers are familiar with will improve their understanding of the design and ultimately realise how a remodelling a credit management system using an Information systems approach can truly benefit them.

6.6 Use of Access Wizards

Access comes with wizards to help with the production of forms and tables. These wizards became very useful for building screen mock ups quickly. These mock-ups (App. D) were then presented to the credit managers who gave feedback on them. With many managers having prior experience with Access it was very easy for them to suggest changes which could be implemented. This I felt was a very important as managers could see how the system was being developed which helps in terms of user acceptance and meeting user's expectations.

A standard template was set out in Access so that all the forms had a familiar feel to them. This ensured that there was a standard feel throughout the system. The design for the screens that's was chosen was the tab folders option to allow easy screen navigation(See user manual for screen shots).The use of buttons was approved for launching queries and performing actions. One of the requests was to make the system as user friendly s possible so to minimise training time and improve usage time.

6.7 Tables

From the ER diagrams tables were designed with field attributes supplied in Access. Access had many pre-defined datatypes which when combined with input formats and rules were enough for most of the fields I needed. In SQL Server the use of User defined data types would have been advantageous but many of datatypes needed were enough for the needs of the prototype.

For the primary key of many fields I opted to choose the incremental autonumber facility provided by Access. This allowed a key to be assigned to data on input. In some cases this could be indexed for faster searches. Too much indexing could result in slower updates which needed to be considered when deciding to index on that field or not.

Validation rules such as “is not null" and selecting table property Required to Yes made sure key fields had to be input on the forms. Screen shots of error messages can be found in User manual.

In this system negative numbers were needed to represent credits therefore rules to allow negative inputs were needed. To confirm an update or save a confirmation box was needed for users to confirm actions before committing them to a database.

6.8 Queries

Queries in Access can either be designed through the query design window (Query by example) or through SQL directly. Another useful feature was the expression builder which helped in getting the input format right for Access. Queries could be used to make filters or forms which were launched on events. Here are two queries used in the system. Others can be found in appendix J.

Ageing Query

```
Invoices1.CustomerID, Invoices1.InvoiceID, Invoices1.InvoiceDate,
Invoices1.PaymentTerms, Invoices1.Amount, Invoices1.Discount,
Invoices1.Payment_date, DateDiff("d",Invoices1!InvoiceDate,Now()) AS [0-30]
FROM Invoices1
WHERE (((Invoices1.InvoiceDate)=Yes) AND ((Invoices1.Paid)=No) AND
((DateDiff("d",[Invoices1]![InvoiceDate],Now()))>0 And
(DateDiff("d",[Invoices1]![InvoiceDate],Now()))<30))
ORDER BY Invoices1.CustomerID, Invoices1.InvoiceID;
```

This query was used to find the difference between when the invoice was put in and the date today. This then worked out how many days the invoice had been unpaid. This was an important feature for credit controllers as they need to chase invoices which have been outstanding past their payment terms.

Fig. 17

Query to get to sum of unpaid

*SELECT DISTINCTROW Customers.CustomerID, Customers.CompanyName, Invoices1.Paid, Sum(Invoices1.Amount) AS [Total Before Discount], Count(*) AS [Number of invoices], Sum(invoices1!Amount-(Invoices1!Amount*Invoices1!Discount)) AS [Balance after discount], Customers.CreditLimit*
FROM Customers INNER JOIN Invoices1 ON Customers.CustomerID=Invoices1.CustomerID
GROUP BY Customers.CustomerID, Customers.CompanyName, Invoices1.Paid, Customers.CreditLimit, Customers.CustomerID
*HAVING (((Invoices1.Paid)=No) And ((Sum(invoices1!Amount-(Invoices1!Amount*Invoices1!Discount)))>Customers!CreditLimit));*

It was challenging to at first get the syntax for Access. It had before the project only been something I had experimented with and I have now learnt a lot from doing this.

Output for Sum Query

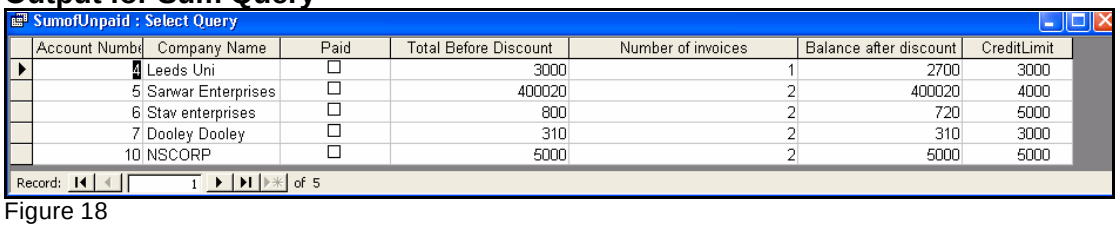

SumofUnpaid : Select Query
Account Numb
Company Name
Paid
Total Before Discount
Number of invoices
Balance after discount
CreditLimit
4 Leeds Uni 3000 1 2700 3000
5 Sarwar Enterprises 400020 2 400020 4000
6 Stav enterprises 800 2 720 5000
7 Dooley Dooley 310 2 310 3000
10 NSCORP 5000 2 5000 5000
Record: 1 of 5

Figure 18

Other queries that I used are included in the appendix J along with some VBA code used to filter forms.

6.9 Security

Security considerations were beyond the scope of the project. The prototype had a system level password for using Access. I felt user level security was not needed as it wasn't part of the requirements of the prototype. If required it would have been implemented.

6.10 Form and Subform Design

Form designs were evaluated by the credit managers before a final form design was chosen. Some of the screen designs can be found in the appendix D which can be compared to Sun screen shots and screen shots of prototype in the user manual.

One of the features most desirable in a system was found to be ease of access to information. Therefore a form was designed where it could display the most relevant information on one screen with links to other information such as invoice ageing. This form and child form relationship is one which is used in Access. It was possible to show customer details and their current balance and how many invoices were outstanding on one form (fig 19). All this information previously would have taken four screens to show.

From here it was possible to further investigate invoice details as well as any calls which were made to pursue the funds. This information is vital for a credit controller when chasing a debt of any kind to establish a case for collection. In any debtors case there could be many calls to one invoice so it was chosen to show calls as a sub form datasheet.

Subforms were particularly useful for displaying queries within the forms. This feature allowed the system to show different details on one screen compared to previous systems where it is important to move to another screen to view this information.

Prototype with customer details and invoice details vs. Sun Systems

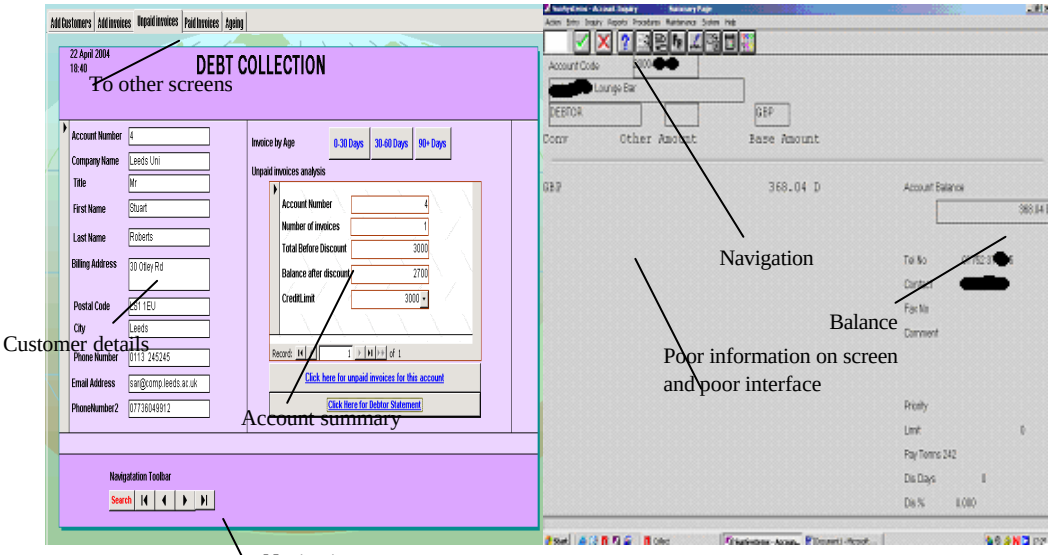


Fig. 19

One of the problems identified in the database issues chapter is the problem of data inconsistency. This can lead to longer search times and incomplete data in the

database. To stop this happening, any fields which needed an input already stored in the database would use a field lookup to get its value. This reduced the amount of inconsistent data in the system. This type of form validation and “is not null” rule for primary keys was used to make sure that data which is needed was entered.

Invoices could only put input after customer details has been entered. One of the major reasons for imposing was this restriction was so that the Invoice belonged to a customer and was not unattached. An editing restriction was put on certain fields to ensure that data could not be changed within the collection screen, and only within customer and invoice screen. Combo boxes also limited the input into the tables by using look up wizards to look up data on other tables, helping normalising issues.

6.11 Reports

Reports were produced and so that they could be printed or emailed to the debtors. The reports formed the Invoices, Account statements and Letters. Examples of which can be found in appendix K. A query was designed to store the details needed on the report. The other fields were taken from the current form loaded on screen.
E.g. to set field in report to form field =[Forms]![index form]![Customers1]![BillingAddress]

To print the invoice to the current account selected a filter was needed. This was then called up in the VBA code to print the current record.
Current record;

```
Dim stDocName As String
stDocName = "Customers2"
DoCmd.OpenReport stDocName, acPreview, "Invoices filter"
```

Filter for unpaid invoices form to show current records details, the form linked to another form running the unpaid query..

```
If Me.NewRecord Then
    Forms![unpaid invoices1].DataEntry = True
Else
    Forms![unpaid invoices1].Filter = "[CustomerID] = " & Me.[CustomerID]
    Forms![unpaid invoices1].FilterOn = True
End If
```

Chapter 7 Testing

7.1 Introduction to Testing

Testing is vital to certify that the system meets the requirements it set out to meet. Bennet, Mcrobb and Farmer (1999) suggest that testing must be done to ensure completeness, correctness, reliability and maintainability. User acceptance testing will be done with people working in the finance sector as well as students who helped in early stages.

White box testing is not really suitable because it tests the structure of the system (loops etc) and the inner workings which is not what I tried to optimise. Instead user acceptance testing and black box testing will be used. Kendall (1992) states that user acceptance testing "Verifies that the system does what it is supposed to do".
Black box testing involves checking to see if the system performs the way it is supposed to. This is very important so that the testers can see how a system such as this could benefit their working practices.

7.2 Database testing

Most of the database testing is done whilst the system was being written. The testing of validation rules and queries can be performed by calculating what the output should be on paper and compare it to system output. Tables can be tested to see if they allow rogue data or out of range data that could affect the system.

7.3 Data entry/Form Testing/ Statements and Letters

Testing of forms to ensure that all fields display the fields they should display. Here tests can be done to make sure that important fields are write-protected. The data on the forms is compared with the table data to check for consistency. Test sets include rogue data, out of range data. The successfulness will depend on the consistency of the input. The statements and letters can be checked against screen data and they can be emailed and printed to make sure they are correct.

7.4 Task based Testing

The main area I am testing is the interface. Therefore I will set the users tasks which need to be accomplished using the system. I will monitor how they use the system and record where I need to give them help. Simple tasks such as adding a customer and adding an invoice will be tested. These tasks will be timed and observed whilst the users use the system. These test forms I have included in appendix F. This type

of testing I felt was more constructive as it demonstrated how useful/useless some aspects of the prototype will be. Although efficiency is an issue, the more important factor I feel is user performance and how well the prototype meets the user's needs. I will be evaluating how first time users of the system can carry out tasks I will set them. To ensure the stability of the system I will perform validation testing as well as destruction testing. Validation testing to ensure that the system does what it should do. Destruction testing will see how it copes when something is there that should not be.

The key things I will monitor are:

- How quickly the task is completed/How much help is needed
- Common pitfalls and user reactions
- How well can they reproduce a task after completing it once.

7.5 System Testing

Form attribute	Expected Result	Actual Result
Customer Form		
Click add record button	Clear Fields for input	Pass
Leave Company name field empty	Error message	Pass
Leave contact name empty	Error message	Pass
Leave billing address empty	Error message	Pass
Leave post code empty	Error message	Pass
Leave telephone number empty	No error message	Pass
Click Save button	Confirm new account option	Pass
Click Clear button	All fields cleared	Pass
Click Delete record	Confirm cascading changes first. If yes then rest is deleted.	Pass
Click Main menu button	Back to main menu	Pass
Invoice Form		
Leave payment terms empty		
Click Save	Confirm update	Pass
Click clear	Clear details	Pass
Click delete	Confirm cascading changes	Pass
Click main menu	Go back to main menu	Pass
Leave amount empty	Cannot be null	Pass
Customer unpaid form		
Correct details displayed	Correct details	Pass
Correct balance displayed	Balance - discount	Pass
Correct number of invoices displayed		Pass
Click Find account	Find function displayed	Pass
Click next record	Next record displayed	Pass
Click Previous Record	Previous record displayed	Pass
Click unpaid invoices for account	Correct invoices are displayed	Pass
Click 0-30 for account	All invoices in that date range for account	Pass
Click 30-60 for account	All invoices in that date range for account	Pass
Click 90+	Invoices above 90 days for account	Pass
Print Debtor Statement/ Letter	Print out of statement	Pass
Email Debtor Statement/Letter	Email statement	Pass
Paid Invoices Form		
Next invoice button	Next invoice	Pass
Previous invoice	Previous invoice	Pass
Click paid invoices	Display paid invoices for current account	Pass
Click Search	Find account number	Pass
Paid Box	Is ticked if paid	Pass
Back to customer button	Goes back to previous form	Pass

Fig. 20

Chapter 8 Evaluation

8.1 What has been achieved and Evaluation of objectives

A working prototype of the system has been developed along with data models of activities. Following the final user testing a feedback from the credit managers and users it was clear the benefits an improved interface could have. The users were able to use the system with minimal help once the basics were introduced.
The issue of convincing managers was tackled through the use fast screen navigation and wealth of information.

The minimum requirements was for screen designs, however this was extended to include a working system with queries and reports. This extra work took a lot of time with changing requirements from managers affecting progress. The issue of SOC support never answering my email hindered any chance of putting a web interface on the system for extra functionality.

The evaluators were impressed how changes to the interface impacted how they used the system, and commented on time possible savings. One manager particularly liked the feature where invoices and statements could be emailed saving time and postage. An auto dialler was integrated if the use of internet telephony was needed. None of these features are in that manager's current system.

Table of objectives I set out in the proposal and how I have met them;

Min. Requirement/ Objective	Status	Evidence
Research into Credit management	Met	Chapter 1
Feasibility study	Met	Chapter 2
Explore current solution	Exceeded	
Carry out interview	Met	Appendix C
Use different modelling tools	Met	UML, Yourdon and Coad DFD, ER, Exceptions
Design prototype screen designs	Met	Appendix D
Test solution	Exceeded	Testing chapter and App F,
Evaluation	Met	Evaluation chap. & app. H
Comment on future	Met	Evaluation
Systems and IS strategy	Met	Analysis chapter
Propose prototype	Exceeded	Database as supplied
Not in Min req. or Objectives but in project		
User manual	Exceeded	Appendix G
System which exceeds minimum requirements set out	Exceeded	System and App. G
Screen outputs	Exceeded	Statements and invoices
Dialler and emailing of invoices	Exceeded	Email an invoice

Fig 21

The test results did show that there was time savings in navigation between screens and users finding information. It was good to see that users could use the system well after a brief introduction. Satisfying some of the usability objectives I set out of

learnability, consistency and re-usability. The rising confidence level of the user was noted. This illustrated how comfortable they felt with the interface.

8.2 Evaluation of prototype

1) Interface-Very positive. Users liked the way they could navigate around the system by just clicking on the tabs. Functional and non functional req. met in system.
2) Usability- Consistency, User friendliness, Re-usability satisfied.
3) Colour scheme- Overall optimistic. Some criticisms of use of colours.
4) Ease of use- Very encouraging comments on ease of navigation, searches and information clarity.
5) Successfulness as a solution- 80% agreed that they could see the benefits of an improved interface. About 20% were confident in their current systems interface.

The prototype had the key features of split screens implemented. This in the current system they are using would have taken 30-40 seconds to navigate to due to more data input needed in Sun. For the colour scheme I opted for mild colours that were a change from the grey screens. However, this wasn't to the taste of everyone.

Screen navigation and information on screen far outweighs Sun used at Company A. Screenshots of the two systems are provided in app. I, so that the difference in design can be seen. Other factors such as database design were only going to be evaluated in terms of efficiency and whether there was a problem in the design which affected the system. Although Sun can manage a lot more data and is more stable, I feel the interface is an improvement. It can however, not be implemented into any company as it is currently. Further comparison of screens can be found in appendix I.

8.3 Further development

1) Development of interface further- Use of split screen such as the demonstration model to give wealth of information.
2) Would need to be rebuilt in SQL server with different front end if to be implemented in any company. Screen designs and manuals were given to all managers who gave feedback to encourage new thinking in new projects.

8.4 Time

The solution was evaluated on time, but it did take up more of my own time which could have been dedicated to other work.

References

Bailey, Robert W. (1989) *Human performance engineering*; Using Human Factors/Ergonomics to achieve computer usability (2nd Edition), Prentice Hall

Bass, R.M. V. (1991) *Credit management* (3rd Edition), Stanley Thornes

Bennet, S., Mcrobb, S., Farmer, R. (1999) *Object Orientated System Analysis and Design using UML*, Mcgraw Hill (1999)

Booth, P.(1981) *An Introduction to Human-Computer Interaction*, Lawrence Erlbarm and associates

Codd. E, (1972) Further normalization of the database relational model

Eason.K.D. (1984) *Towards experimental study of usability*; behaviour of IT

Edwards, B. (1997) Credit management Handbook (4th edition)

Flynn, Donal J. (1992) *Information systems requirements*; Determination and analysis, Mcgraw-Hill

Lingaard, G. (1994) *Usability testing and system evaluation;* A guide for designing a useful computer system (1994)

Kendall,P.(1992) Introduction to Systems Analsyis and design: A structured approach (2nd edition) WM.C Brown

Kendall, K., Kendall, J. (1999) Systems analysis and Design (4th edition) Prentice Hall

McConnell, S. (1996) Rapid development : taming wild software schedules, Microsoft Press

Priestley, M. (2000) Practical object-oriented design with UML , McGraw-Hill

Sauer, C. (1993) *Why information systems fail*; A case study approach, Information system series

Schneiderman, Ben., Designing the user interface; strategies for effective Human-Computer interaction (2nd Edition) Addison-Wesley (1992)

Shackel, B. (1986) *Ergonomics in design for usability*, In people and computers; Design, Human-Computer specialist group, British Computer society, Cambridge University Press.

Shackel, B, (1991b) *Human factors for information Usability*; Background and overview in Human factors for Informatics usability, Cambridge university press

Journals
Rambaugh, J., “Models through the development process”, Journal of object orientated programming, May 1997

McFarlan,F.W., “Information technology changes the way you compete”, Harvard Business School Review, May-June 1984, 93-103

Ives.B, Learmonth, G.P.,”The Information systems as a competitive weapon”, Communications of the ACM, Vol. 27 No.12, 1984, 1193-1201

Galliers, R.D, “Information systems and technology planning within a competitive strategy framework’, The Role of Information Management in Competitive Success, Paergamon Infotech State of the Art Report, 1987

Online Resources

www.icm.org.uk – Institute of credit management website

www.utteraccess.com – Access help

From the onset of this project I had a vision of how the system would turn out. From doing this project I was hoping to put al my knowledge I had acquired into something which was tangible. But what I didn't realise was the scope of it all. The effects of lost work, database not working and other code errors all put me back on my schedule. The objectives I had defined at the start of the project were in hind sight rather ambitious and only blood, sweat and tears got me to meet them.

Advice to students who use professionals in their research is to have extra time set aside. Finding a time which suits you both can take a long time. Also be realistic in who you attempt to contact for insight. I felt contacting the credit management research department at Leeds University Business School would benefit my research. It turned out fruitless. It is amazing to see just how little people in the real world want to help you. Following lots of unreturned emails and phone calls the remaining testers of the system had to be students. In all I got to have 2 managers and 2 credit controllers to test the system. The rest I had to enlist the aid of students to help me through.

Another reflection is to always have in the back of your mind what the entire scope of your project is. In my case the requirements changed every time I spoke to a credit manager. From just screen designs to a system, to emailing reports and statements.
Although I am still bitter about SOC support never returning my email I found that using Access did have one advantage. That was that I didn't need to travel 200 miles home to demonstrate early stages of the system. These could be emailed and therefore cut out the costs of travelling. In future this will remind me to use a platform that the test sites already have.

The project progress in some areas was realistic although delays in replies and other problems caused me to fall behind. An underestimation on my part was just what a 20 credit project involved. I had projects which were forty credits, so naturally I did the same amount of work and found myself deleting 15 pages. The 20 credit project has in fact been harder and more time consuming than any other 20 credit module I have ever done.

I am hoping that the time I have dedicated to this does not affect my other modules as I have neglected them to pursue this. Once the system starts working its hard to leave alone!

One area of concern I had was to why I was given a supervisor whose area of expertise was not really in the field of the project I was doing. In all fairness, my supervisor was amazing in the help she gave me but sometimes I felt I could have benefited from a more business orientated supervisor. In future I would suggest that if people do propose their own projects, and then make sure your supervisor knows what you are doing.

Another lesson to add to my collection is that to always use naming conventions when programming. Trying to remember what command51 was

can be a challenge. This is why during the last stages buttons and other attributes had more identifiable names.

For all the non programmers who do IS it is advisable to get some stress balls when doing any project where coding is needed. The database at times could freeze and not even tell me what the error was (Thank you Microsoft). For a long time printing an invoice would cause the project to crash. Which apart from being annoying was also starting to impact my database? Eventually this was fixed but 4 hours lost on one problem.

Changing requirements was an issue to. Three days ago I got a call from a credit manager to ask if I could change her previous request of the amount to be split into quantity and price. This I had to tell her was too late and the prototype could demonstrate the amount just as well without the quantity and price.

It has however not all been bad. This project has become the fruit of my computing studies and is something I can be proud of.

My advice to anyone doing a project in this area
1) Professionals take time in replying; have buffers for this extra time.
2) Use meaningful names when coding
3) Make back up copies all the time, one in every room if needs be.
4) Don't be optimistic as to how much help you can get. Be realistic.
5) Editing and re-writing takes up time-prepare for this.
6) Carefully think about your minimum requirements and objectives.
7) Carefully plan your time.
8) Prepare for changing requirements.
9) Finally set the scope of the project. Aim for this, surpassing is good, but don't neglect your other modules.

Gannt Chart

Week Number

Stages		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Project Idea																											
Project Proposal																											
Scope and tasks making up project																											
Estimated time to be taken on each stage																											
Assumptions and constraints -																											
Background research																											
Mid project Report																											
Feasibility Research																											
Requirements analysis																											
Analysis of current solutions																											
Modelling of new solution																											
Implementation																											
Delivered solution																											
Testing of solution																											
Evaluation of solution																											
Review of stage																											
Report Writing																											
Week 1	WE 19-sept 2003																										
Week 5	WE 17-Oct 2003																										
Week 10	WE 12-Dec 2003																										
Week 15	WE 16 Jan 2004																										
Week 17	WE 20-Feb 2004																										
Week 19	WE 5 Mar 2004																										
Week 26	WE 23 April 2004																										

What systems have you worked on, can you comment on how successfully they worked?

The two main systems I have worked on are Sun and Oracle. Both systems work well if companies use them to the limit they were originally supplied for. However as a business grows a system needs to grow as well, this may mean implementing new systems or upgrading old ones in line with the needs of a growing company.

In your opinion what do you perceive as the most important factor in a system designed to aid credit manager/controllers? Do factors such as the ability to look up customer details, or account ageing effect your job or your employee's jobs?

For a credit controller's perspective easy access to account enquiry, address, invoice details and credit terms are the main factors for any good in house system to work well and be productive.

In the systems you have worked with what would you say has been the major pitfalls in them and how would you improve them?

The major pitfalls in systems I have worked with is the over loading of the system. Upgrading and implementing new systems would improve this. Some systems also need to have split screens so all enquiries can be seen on one screen instead of numerous screens.

Do you find current systems complex and sometimes excessive to the needs of the employees? Would you prefer a smaller more tailored version?

Yes, Oracle can be very mind blowing and would be more user friendly if it was slightly tailored.

What sort of features do you think are 1) Vital for the systems success? 2) Provide use for the goals of your team? 3) Have no use for you at all?

1. Split screens
2. Speed and information
3. Duplicate screens

If you had the opportunity to design a system using your experience and knowledge of credit management area, how would you design the system so that it was tailored to the real work environment?

Split screens, systems that can raise credits on a daily basis showing on the screen on the same day.

What do you think your credit controllers struggle most with when they are using the systems?

Having to keep coming out of one part of the system and going into another part of the system.

What goals would you use to measure the success of a system?

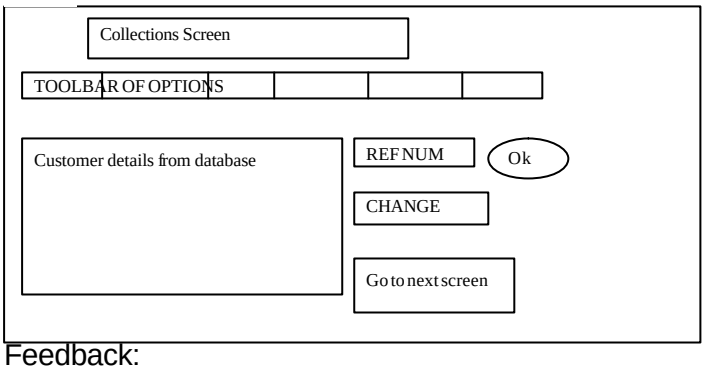
Speed and capability to hold the required information.

How would you like to see these goals implemented?

Trail runs of different systems

Anything else you would like to add regarding the systems you have used in the real work environment?

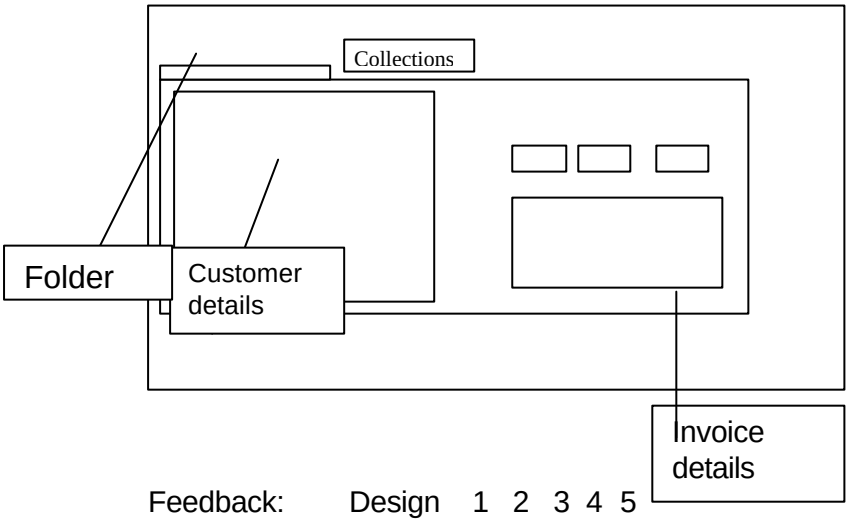
1)



Feedback:

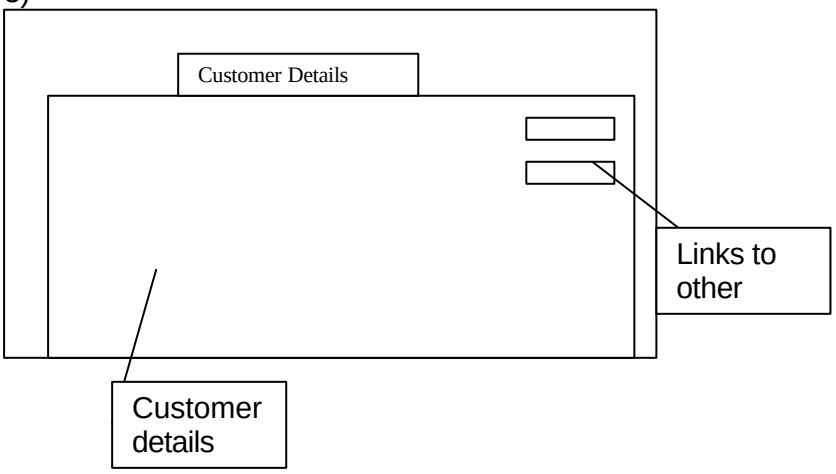
Design 1 2 3 4 5
Clarity 1 2 3 4 5

2)



Feedback: Design 1 2 3 4 5
Clarity 1 2 3 4 5

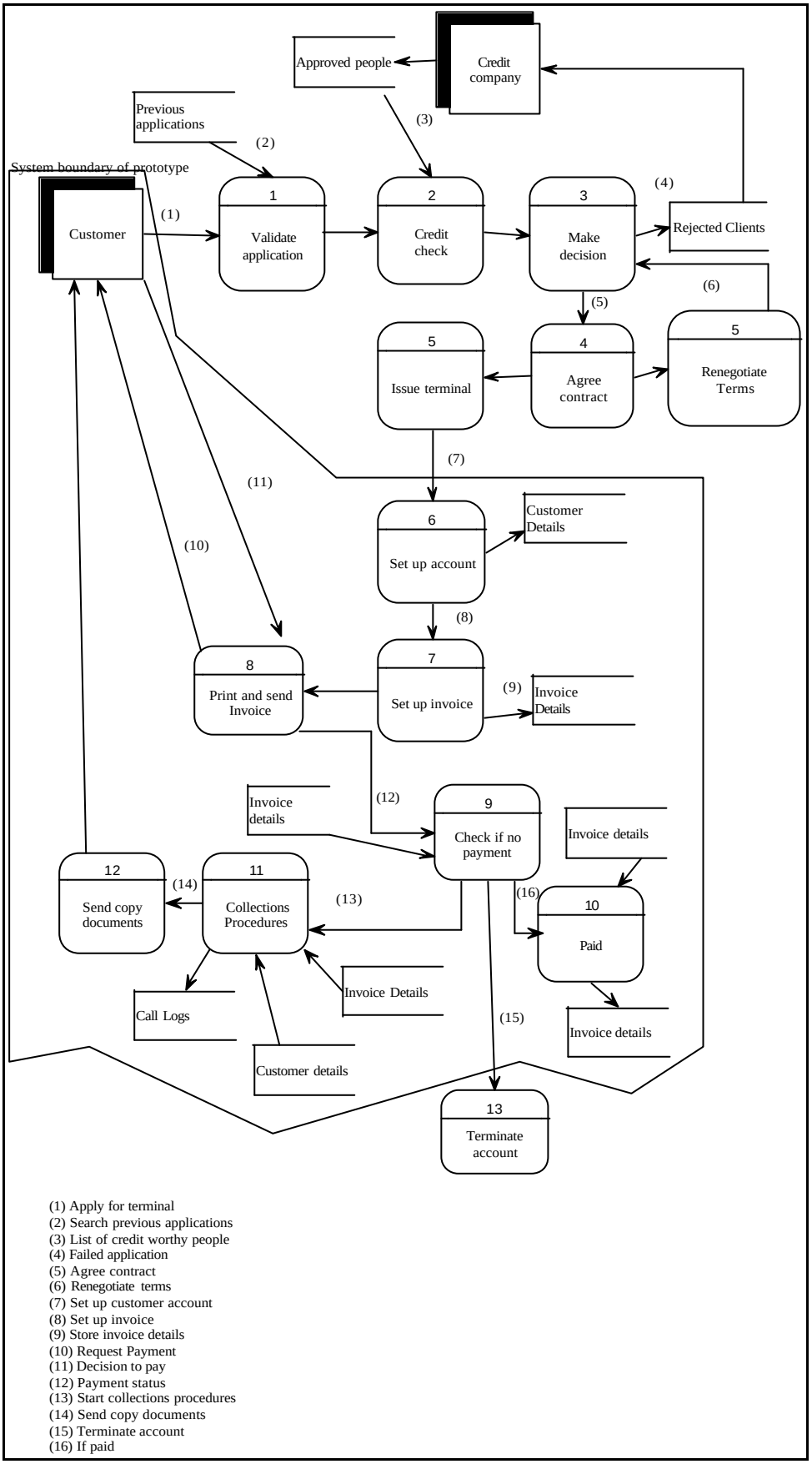
3)



Feedback: Design 1 2 3 4 5
Clarity 1 2 3 4 5

Which layout do you prefer 1 2 3

Comments:


Approved people
Credit company
Previous applications
(3)
(2)
System boundary of prototype
Customer
(1)
1
Validate application
2
Credit check
3
Make decision
(4)
Rejected Clients
(6)
(5)
5
Issue terminal
4
Agree contract
5
Renegotiate Terms
(7)
(11)
6
Set up account
Customer Details
(10)
(8)
8
Print and send Invoice
7
Set up invoice
(9)
Invoice Details
Invoice details
(12)
9
Check if no payment
Invoice details
12
Send copy documents
(14)
11
Collections Procedures
(13)
(16)
10
Paid
Invoice Details
Call Logs
(15)
Invoice details
Customer details
13
Terminate account
(1) Apply for terminal
(2) Search previous applications
(3) List of credit worthy people
(4) Failed application
(5) Agree contract
(6) Renegotiate terms
(7) Set up customer account
(8) Set up invoice
(9) Store invoice details
(10) Request Payment
(11) Decision to pay
(12) Payment status
(13) Start collections procedures
(14) Send copy documents
(15) Terminate account
(16) If paid

Exercise	Result	Time Taken	Help needed
Add a customer with the details Company Name: Tom Barnes company Title: Mr Full name: Tom Barnes Address: 21 Trescin Place Post Code: LS6 1EP City: Leeds Phone number: 01132 245245 Phone number 01132 245246 Email Address: tom.barnes@email.com Credit limit: 4000			
Add an invoice with the details Customer ID : 5 Invoice Date : today Payment Terms: 30 day Discount: 10% Amount:3500 Currency: GBP Terminal Type:T7P Paid: No Payment Date: none Invoice Type: debit Credit note for: n/a Notes: Problems paying last time			
Search for customer with name “Sar*”			
Go to next record			
Go to previous record			
What is the balance on account with ID “5”			
What invoices relate to this account			
How many calls were made on invoice number “5”			
Go back to customer form			
What invoices are 0-30 days old			
What invoices are 30-60 days old			
What invoices are 60-90 days old			
What invoices have been paid by customer with ID “5”			
Search for customer with invoice number “6” in paid invoices			

Exercise	Result	Time Taken	Help needed
Add a customer with the details Company Name: Tom Barnes company Title: Mr Full name: Tom Barnes Address: 21 Trescin Place Post Code: LS6 1EP City: Leeds Phone number: 01132 245245 Phone number 01132 245246 Email Address: tom.barnes@email.com Credit limit: 4000			
Add an invoice with the details Customer ID : 5 Invoice Date : today Payment Terms: 30 day Discount: 10% Amount:3500 Currency: GBP Terminal Type:T7P Paid: No Payment Date: none Invoice Type: debit Credit note for: n/a Notes: Problems paying last time			
Search for customer with name “Sar*”			
Go to next record			
Go to previous record			
What is the balance on account with ID “5”			
What invoices relate to this account			
How many calls were made on invoice number “5”			
Go back to customer form			
What invoices are 0-30 days old			
What invoices are 30-60 days old			
What invoices are 60-90 days old			
What invoices have been paid by customer with ID “5”			
Search for customer with invoice number “6” in paid invoices			

USER MANUAL
Prototype CM System

Taking you from here To here

User manual as distributed during testing

CONTENTS

1.0……Introduction
2.0 ……Starting the system
2.1 ……Introduction to the interface
3.0……Time to get started
3.1……Adding a customer
3.2 ……To search for a customer
3.3 ……To delete a customer
4.0 ……To add an invoice
4.1 ……To delete an invoice
4.2 ……To Print Invoice
4.3 ……To Email Invoice
4.4 ……Search for a customer
5.0 ……What is on the Collections screen
5.1 ……What can I do on this screen
5.2 ……How to view all unpaid invoices for an account
5.3 ……View ageing of individual invoice.
5.4 ……Print/Email debtor statement
6.0 ……Search for paid invoices
7.0 ……Search for all unpaid invoice ageing
8.0 ……Error messages
9.0 ……Final Word

1.0 Introduction

Welcome to the manual for the prototype credit management system. This system will demonstrate the new and improved way of working in credit control. This manual will provide a step by step guide on how to use the system and how it works. If you have any problems using the system please do not hesitate to contact me on Sarwar_Imran@hotmail.com.

I feel the best way to learn a system is to use it. So this is provided for helping you when using it for the first time. Good luck!

2.0 Starting the system

To start the database you must first ensure you have

- Microsoft Access
- Microsoft Word

Once you have these minimum requirements please open up the database called Fyp1.mdb. This will bring you to a switchboard.

Switchboard

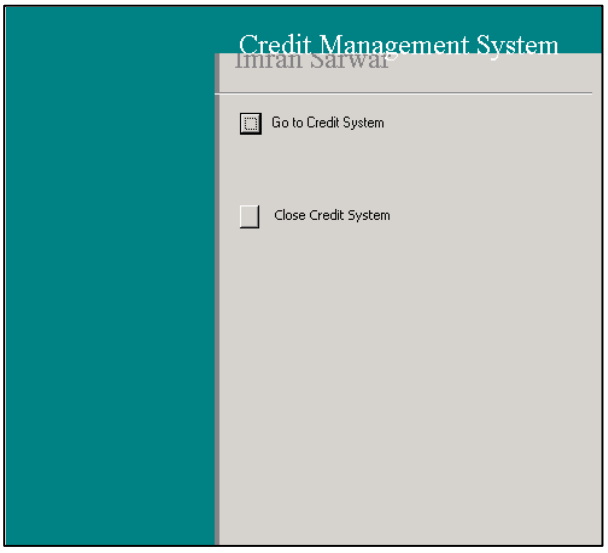


Figure 1.0

Please select the credit management system option and you are then ready to begin.

2.1 Introduction to the interface

To select different forms the tab folders can be used. This is to ensure fast switching in between screens.

Figure 1.1

To navigate to records the navigation toolbar is provided

Figure 1.2

Pressing the buttons will perform the actions as written on the buttons. In the system these are used to launch queries, reports and forms.

3.0 Time to get started

3.1 Adding a customer

First of all in order to set up the database a customer is needed. The screen below is the customer details screen. Here the customer details can be input into the system.

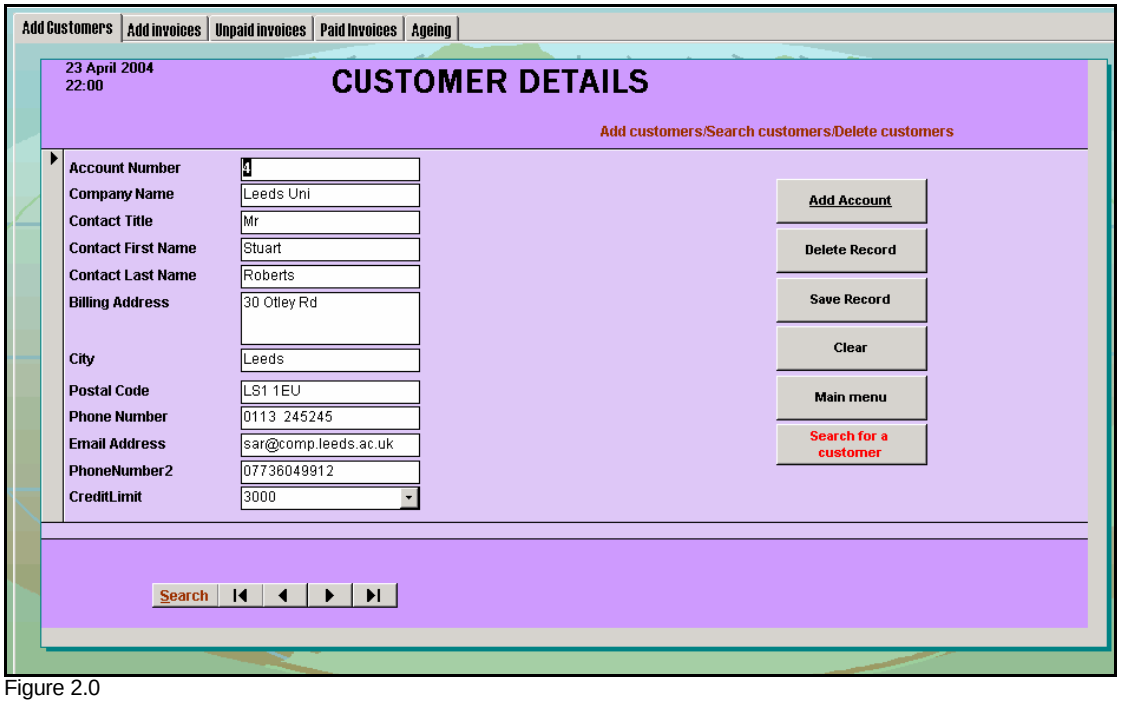


Figure 2.0

To add a customer you have to:

1) Press Add new record
2) Fill in the details
3) Rules for the following fields exist. These therefore cannot be violated;
    - Company name- cannot be null
    - Account number –cannot be null (automatically generated)
    - Contact name –cannot be null
    - Billing address –cannot be null
    - Post code –Cannot be null
    - City- cannot be null

4) Once the details have been entered press the save button to save the record.

A confirm box will appear. Click yes.

5) This customer has now been generated.

Pressing clear at any time will clear the details you have entered.

Main menu option will take you back to the switch board.

3.2 To search for a customer

1) Select the field in which you want to search within.
2) Click the search button.
3) Wildcard queries such as Im* will return Imran, Imy, but not Tim.

3.3 To delete a customer

1) Navigate to the customer and press delete.

Note: Customer cannot be deleted if there are invoices still attached to them.

4.0 To add an invoice

1) Press add invoice
2) Fill in the details as before (default date is today and Invoice ID generated automatically.
3) Relate invoice to customer by selecting Customer ID from the drop down box.
4) Select terminal type
5) Leave Payment date blank. This is filled in when customer has paid
6) Select Invoice type from drop down box. Debit for an invoice. Credit for a credit note.
7) If credit note is selected then fill in the invoice number to which the credit note relates to.
8)

9) Any special invoice notes can be input into the box

10) Press Save. A confirm box will appear. Press Yes

Note: Pressing clear will clear any data input onto screen.

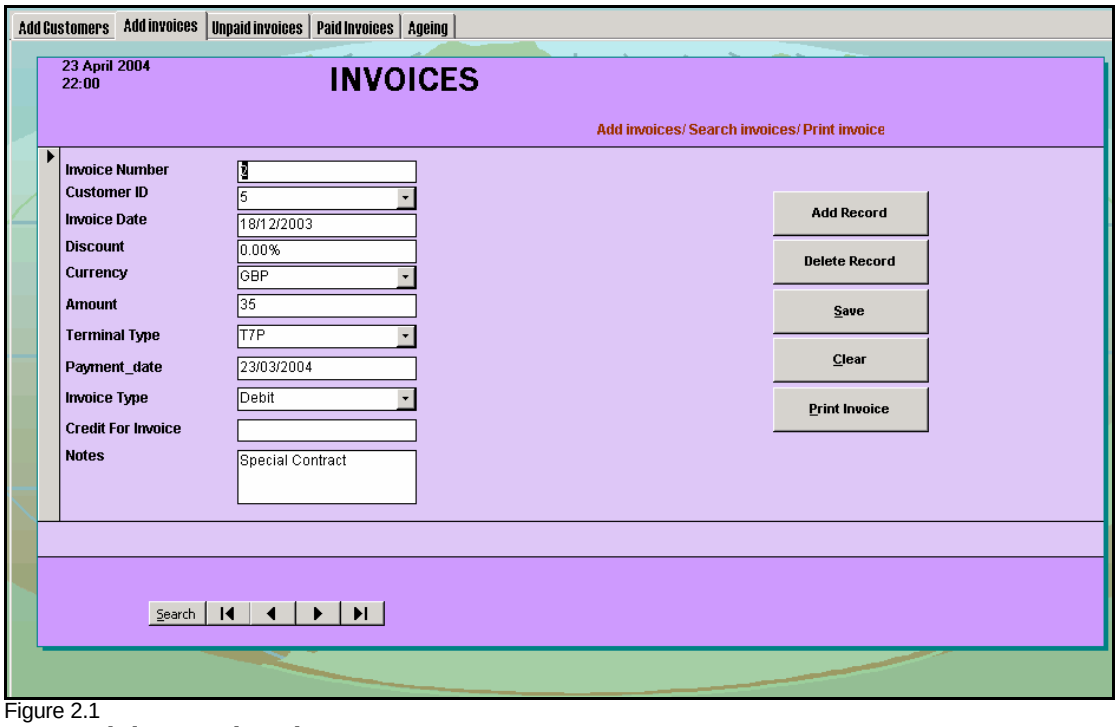


Figure 2.1

4.1 To delete an invoice

1) Navigate to invoice and press delete

Note: Invoice cannot be deleted if it impacts cascading changes.

4.2 To Print Invoice

1) Select the invoice you wish to print.
2) Press the print button.
3) A preview box will appear (fig 2.2). If invoice is correct then press the print icon in left hand corner or right click->Print.

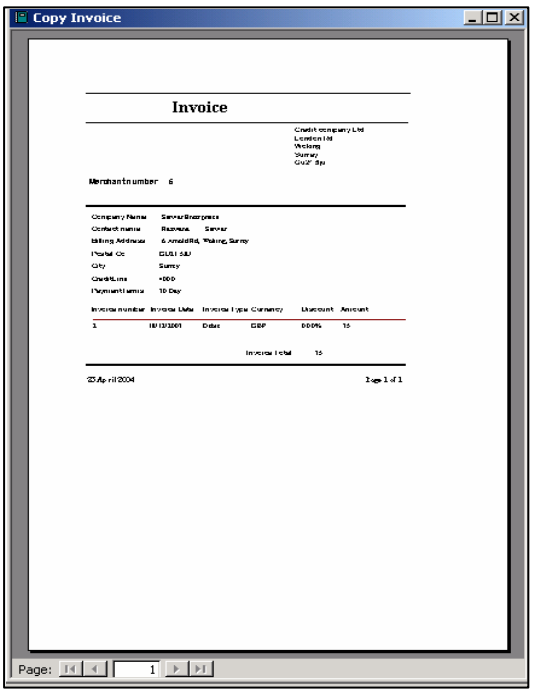


Figure 2.2

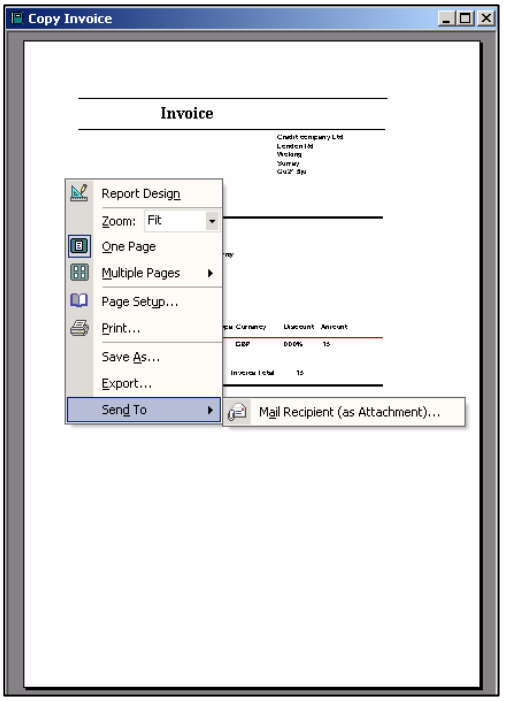


Figure 2.3

4.3 To Email Invoice

1) Select the invoice you wish to email
2) Press the print invoice button
3) On the preview screen right click the mouse (fig 2.3).
4) An option to email should be present. Select this option.
5) This will open up the default mail client and attach the invoice to it.

Note: The file is attached as a .tmp. When downloading the file change extension to .doc and the invoice will open up in Word or any other word processor (Fig 2.3). This feature is not currently in Sun systems that are in use.

4.4 Search for a customer

1) Select the field in which you want to search within.
2) Click the search button.
3) Wildcard queries such as Im* will return Imran, Imy, but not Tim.

5.0 What is on the Collections screen?

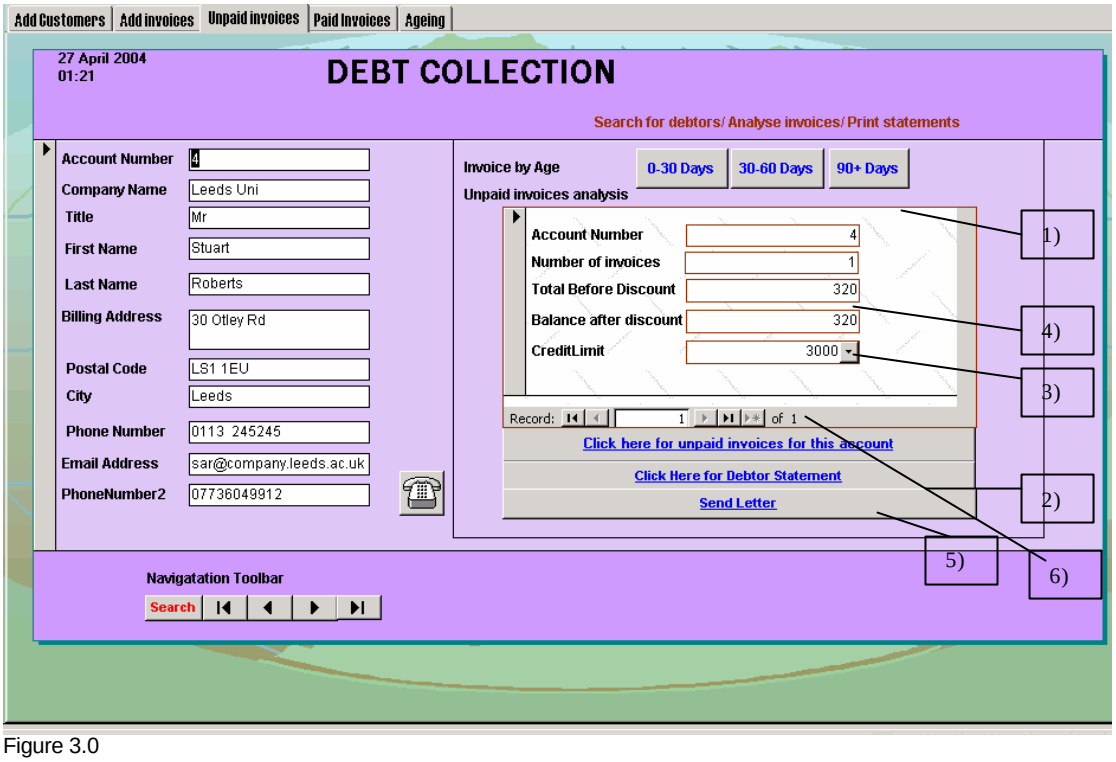


Figure 3.0

5.1 What can I do on this screen (fig 3.0)

1) View ageing of individual invoices
2) View all unpaid invoices
3) View the running account balance, before and after discount.
4) View the running total number of unpaid invoices
5) Print a debtor statement
6) Email a debtor statement

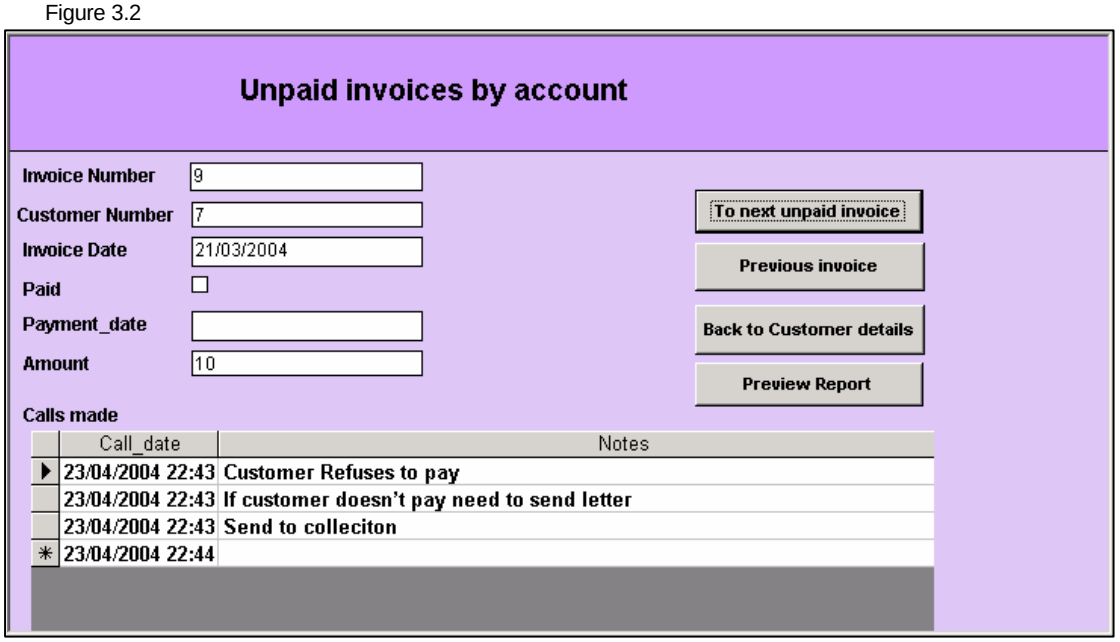
5.2 How to view all unpaid invoices for an account

1) Select an account
2) Click the “View all unpaid invoices for this account” button.

Figure 3.1

3) In here the calls made to the debtor have been logged.
4) The buttons will navigate to the next record if it exists.
5) Clicking the back to customer details screen will close the form.

Figure 3.2



5.3 View ageing of individual invoice.

1) Select an account
2) Press either 0-30 Days, 30-60 Days or 90+ day's buttons for corresponding invoices.
3) From there screens if there are more than one invoice a navigation bar will appear.

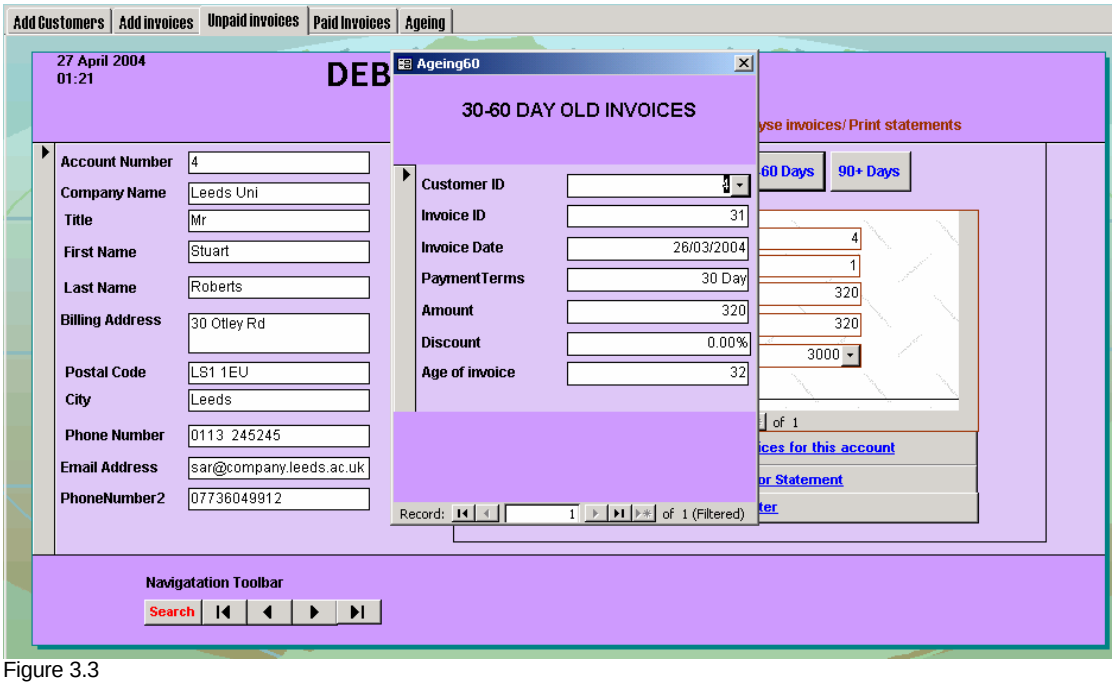


Figure 3.3

5.4 Print/Email debtor statement

1) Navigate to record
2) Press the "Click here for Debtor statement" button.
3) A preview will appear
4) To print press the printer icon or right click and select print.

5) To email the report right click on the preview and select email as an attachment. (Same as 4.3).

6.0 Search for paid invoices

1) Select Paid invoices.
2) Navigate to account number using search option of navigation toolbar.
3) Press “Paid invoices” for a list of all paid invoices for that account to come up.
4) To close press “Back to customer details”.

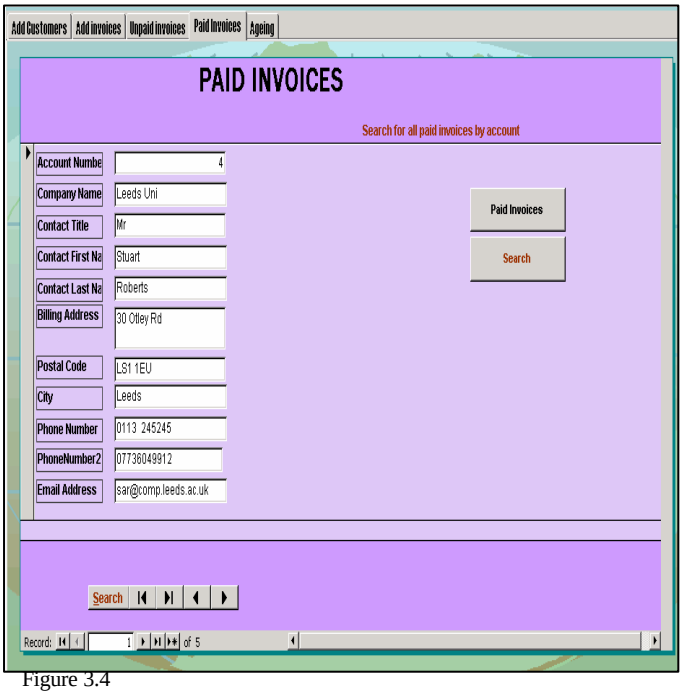


Figure 3.4

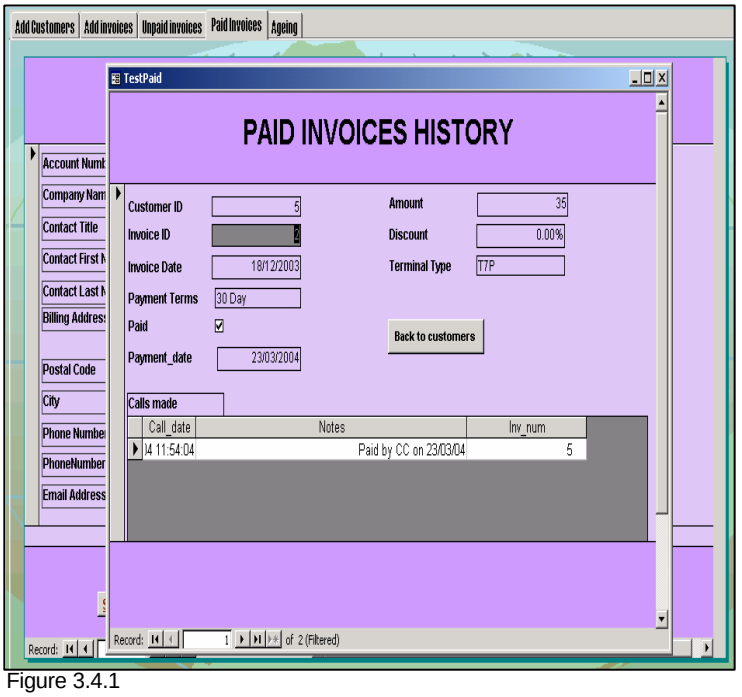


Figure 3.4.1

7.0 Search for all unpaid invoice ageing

1) Navigate to ageing screen
2) Select a time period by clicking the corresponding button in fig 3.5.
3) The number of invoices unpaid in that date range will now appear.

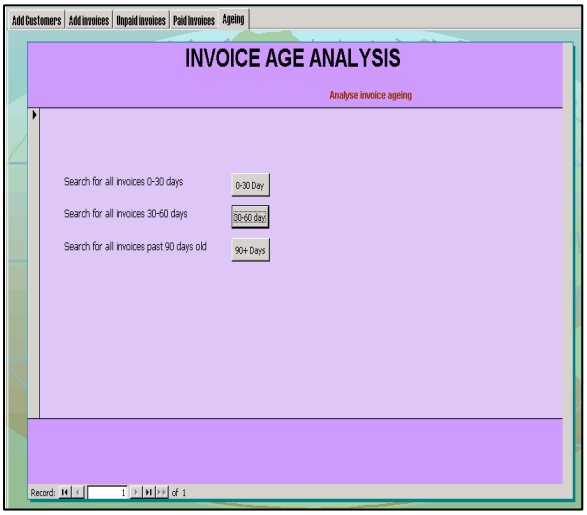


Figure 3.5

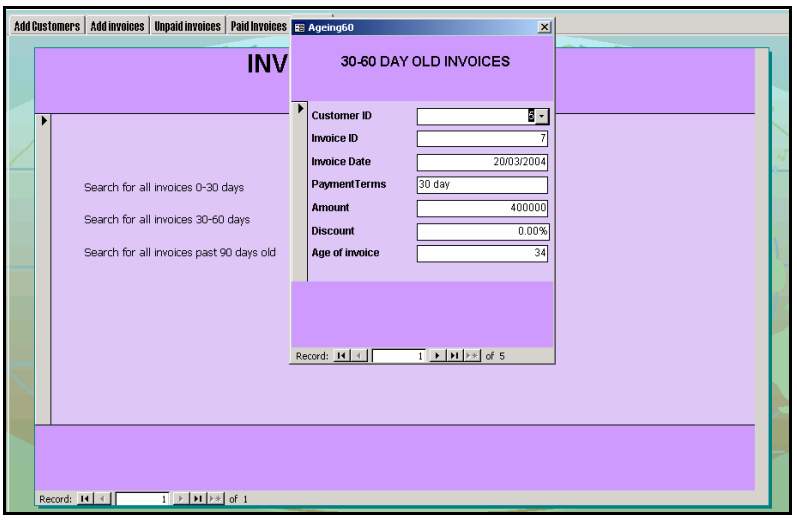
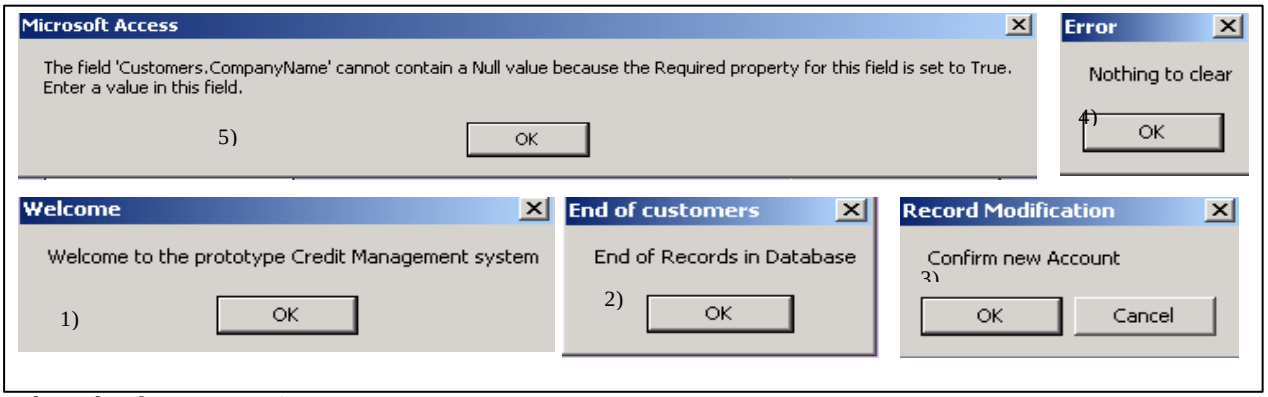


Figure 3.5.1

8.0 Error messages



What do they mean?

Figure 4.0

1) Welcome screen for system

2) This message signifies that there are no more records to navigate to.

3) This message is a confirmation of a save request. Pressing ok will save the account and pressing cancel with not save the record.

4) If the clear button is pressed to many times this message will appear. It is a message to say that there is nothing left to clear on the form.

5) This messages appear on saving an invoice or customer without first filling in the required fields.

9.0 Final Word

Well that's it for the user manual. Please feel free to explore the system and if you have any questions to contact me. Your help has been important in trying to get this solution to meet your needs. So many thanks.

Your help is just now required to fill in the evaluation questionnaires and test the system. I hope this shows you a better way of using your information systems capabilities.

Imran Sarwar

Evaluation Form for Prototype Appendix H

1 very easy 2 quite easy 3 moderate 4 difficult 5 very difficult

1) Rate ease of use 1 2 3 4 5

2) Rate information wealth High/Medium/Low

3) How did you find adding a customer 1 2 3 4 5

4) How did you find adding a invoice 1 2 3 4 5

5) Did you have trouble locating the account balance Yes/ No

6) Rating of interface Excellent /good/ ok/ poor/ very poor

7) Rating of colour scheme Excellent/ good/ ok/ poor/ very poor

8) Is this system easier to use than your current system Yes/No

9) Would your work take less time to do with this sort of interface Yes/No

10) Does the prototype meet your requirements from a system Yes/No

10) Any problems

……
……
……

11) Suggestions for future improvements

……
……
……

Signed ……. Date ……

Evaluation Form for Prototype Appendix H

1 very easy 2 quite easy 3 moderate 4 difficult 5 very difficult

1) Rate ease of use 1 2 3 4 5

2) Rate information wealth High/Medium/Low

3) How did you find adding a customer 1 2 3 4 5

4) How did you find adding a invoice 1 2 3 4 5

5) Did you have trouble locating the account balance Yes/ No

6) Rating of interface Excellent /good/ ok/ poor/ very poor

7) Rating of colour scheme Excellent/ good/ ok/ poor/ very poor

8) Is this system easier to use than your current system Yes/No

9) Would your work take less time to do with this sort of interface Yes/No

10) Does the prototype meet your requirements from a system Yes/No

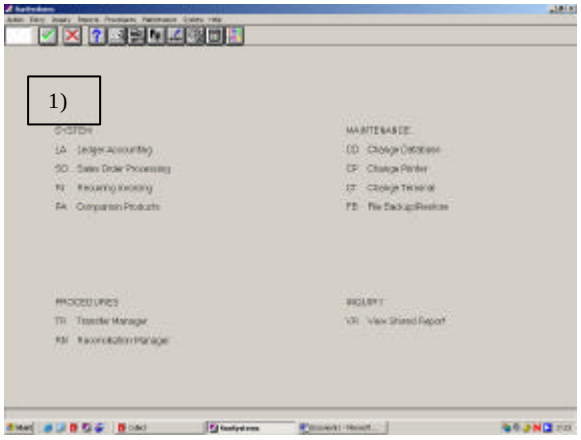
10) Any problems

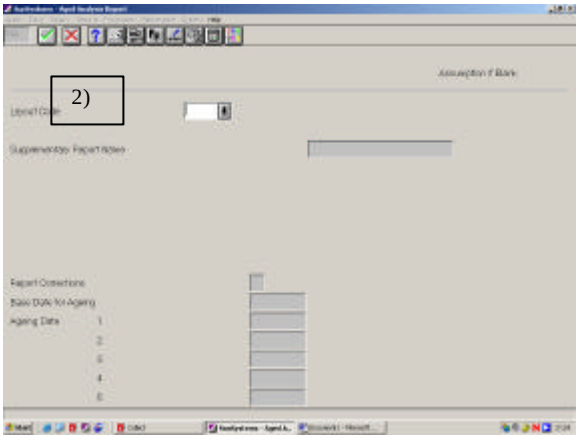
……
……
……

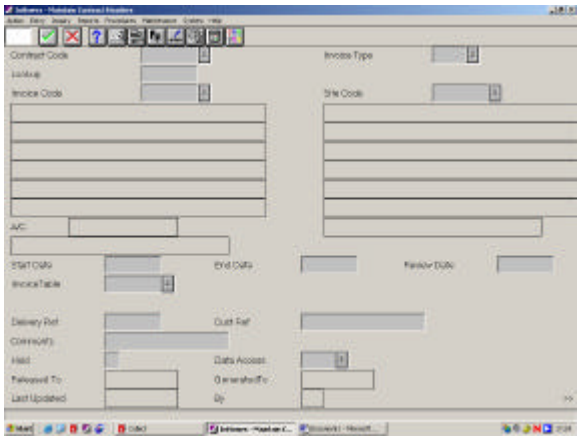
11) Suggestions for future improvements

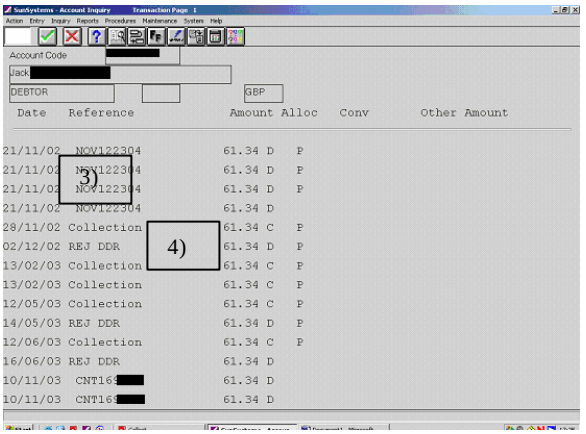
……
……
……

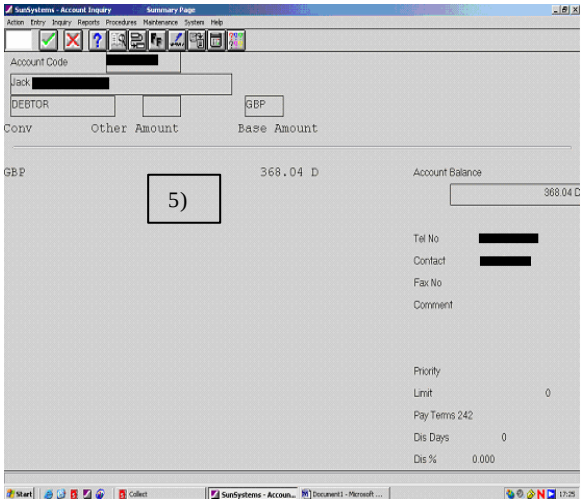
Signed ……. Date ……

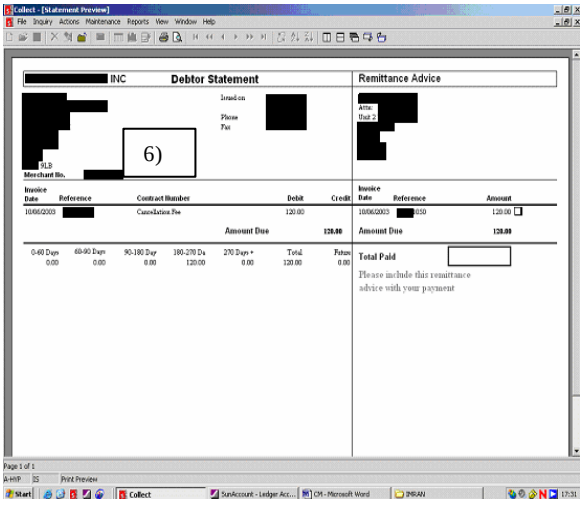






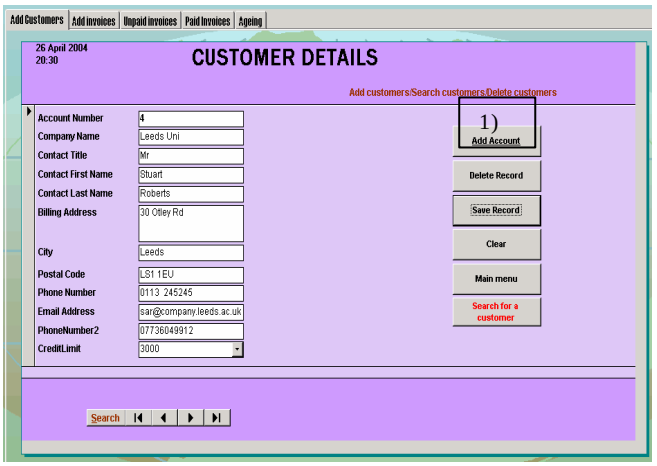


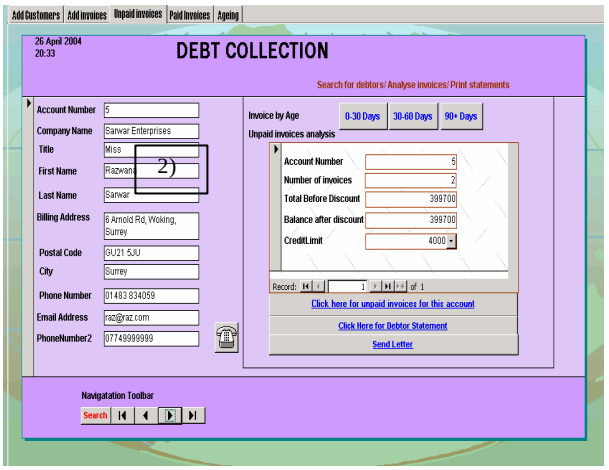


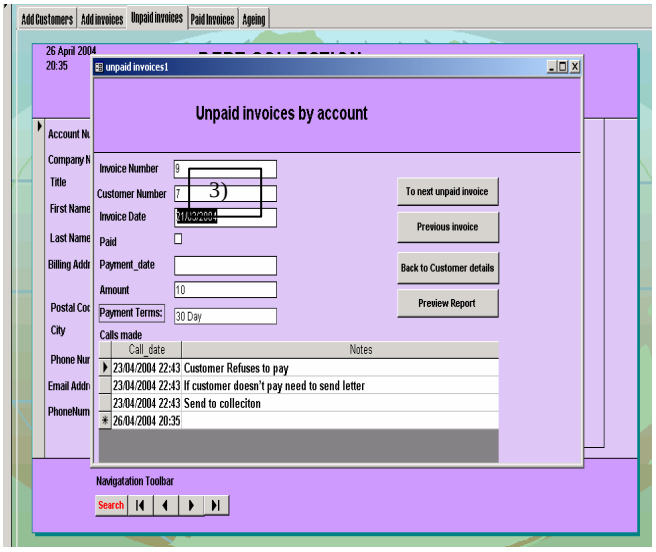


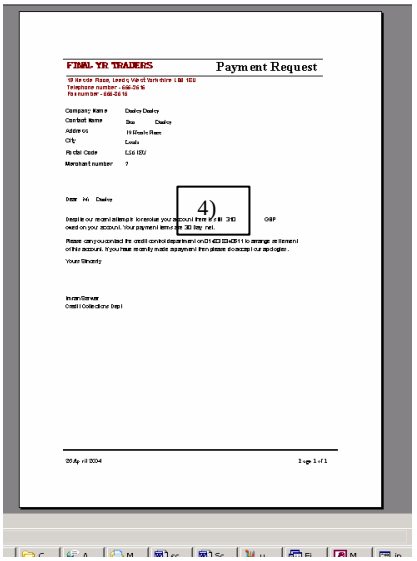
1) Menu Screen
2) Invoice details
3) Adding customer details
4) Customer breakdown
5) Customer details screen
6) Statement

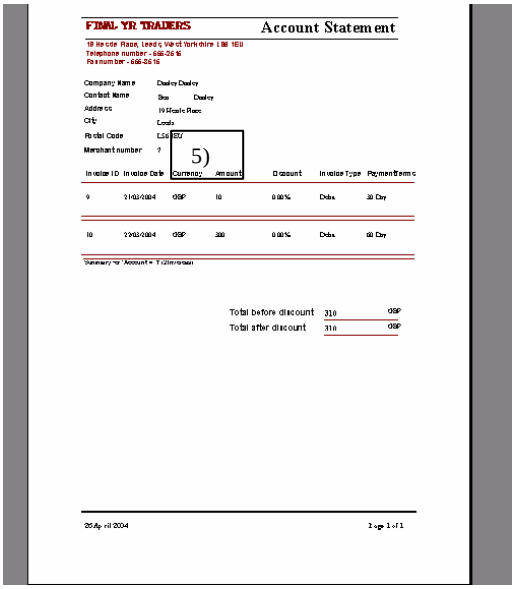
Prototype











1) Add customer details
2) Collection screen with split screen account breakdown
3) Unpaid invoices for each account
4) Payment letter that could be emailed or printed
5) Account statement

Query for find debtors unpaid 30-60 days

```
SELECT Invoices1.CustomerID, Invoices1.InvoiceID, Invoices1.InvoiceDate,
Invoices1.PaymentTerms AS Expr1, Invoices1.Amount, Invoices1.Discount,
Invoices1.[Terminal Type], Invoices1.Paid, Invoices1.Payment_date,
DateDiff("d",Invoices1!InvoiceDate,Now()) AS [60-90], Invoices1.PaymentTerms
FROM Invoices1
WHERE (((Invoices1.InvoiceDate)=Yes) And ((Invoices1.Paid)=No) And
((DateDiff("d",Invoices1!InvoiceDate,Now()))>30 And
(DateDiff("d",Invoices1!InvoiceDate,Now()))<60))
ORDER BY Invoices1.CustomerID, Invoices1.InvoiceID;
```

Query to calculate sum of unpaid invoices

```
SELECT DISTINCTROW Customers.CustomerID, Customers.CompanyName,
Invoices1.Paid, Sum(Invoices1.Amount) AS [Total Before Discount], Count(*) AS
[Number of invoices], Sum(invoices1!Amount-
(Invoices1!Amount*Invoices1!Discount)) AS [Balance after discount],
Customers.CreditLimit
FROM Customers INNER JOIN Invoices1 ON Customers.CustomerID =
Invoices1.CustomerID
GROUP BY Customers.CustomerID, Customers.CompanyName, Invoices1.Paid,
Customers.CreditLimit, Customers.CustomerID
HAVING (((Invoices1.Paid)=No));
```

Finding Paid invoices

```
SELECT Invoices1.CustomerID, Invoices1.InvoiceID, Invoices1.InvoiceDate,
Invoices1.PaymentTerms, Invoices1.Amount, Invoices1.Discount,
Invoices1.[Terminal Type], Invoices1.Paid, Invoices1.Payment_date
FROM Invoices1
WHERE (((Invoices1.InvoiceDate)=Yes) AND ((Invoices1.Paid)=Yes))
ORDER BY Invoices1.CustomerID, Invoices1.InvoiceID;
```

Exceed credit limit

```
SELECT DISTINCTROW Customers.CustomerID, Customers.CompanyName,
Customers.CreditLimit, Sum(Invoices1.Amount) AS [Sum Of Amount], Count(*) AS
[Count Of Invoices1], Invoices1.Paid
FROM Customers INNER JOIN Invoices1 ON Customers.CustomerID =
Invoices1.CustomerID
GROUP BY Customers.CustomerID, Customers.CompanyName,
Customers.CreditLimit, Invoices1.Paid
HAVING (((Sum(Invoices1.Amount))>[customers].[creditlimit]) AND
((Invoices1.Paid)=No));
```

Useful VBA

Filtering a form

```
Private Sub FilterChildForm()

    If Me.NewRecord Then
       Forms![TestPaid].DataEntry = True
    Else
       Forms![TestPaid].Filter = "[CustomerID] = " & Me.[CustomerID]
       Forms![TestPaid].FilterOn = True
    End If

End Sub
```

More meaningful error messages

```
Err_Command29_Click:
    MsgBox "End of Records in Database", vbOKOnly, "End of Invocies"
    Resume Exit_Command29_Click
```

Opening form on current record

```
    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "Ageing30"

    stLinkCriteria = "[CustomerID]=" & Me![CustomerID]
    DoCmd.OpenForm stDocName, , , stLinkCriteria
```

Filter Preview to current record

```
    Dim stDocName As String

    stDocName = "Customers2"
    DoCmd.OpenReport stDocName, acPreview, "Invoices filter"
```

Appendix K
Letters and Statements as printed from the prototype.
These can be emailed/edited or printed

FINAL YR TRADERS

Account Statement

19 Hessle Place, Leeds, West Yorkshire LS6 1EU
Telephone number - 555-2515
Fax number - 555-3515

Company Name	Leeds Uni
Contact Name	Stuart Roberts
Address	30 Otley Rd
City	Leeds
Postal Code	LS1 1EU
Merchant number	4

Invoice ID	Invoice Date	Currency	Amount	Discount	Invoice Type	PaymentTerms
31	26/03/2004	GBP	320	0.00%	Debit	30 Day

Summary for 'Account' = 4 (1 Invoice)

Total before discount	**320**	GBP
Total after discount	**320**	GBP

FINAL YR TRADERS

Payment Request

19 Hessle Place, Leeds, West Yorkshire LS6 1EU
Telephone number - 555-2515
Fax number - 555-3515

Company Name	Leeds Uni
Contact Name	Stuart Roberts
Address	30 Otley Rd
City	Leeds
Postal Code	LS1 1EU
Merchant number	4

Dear Mr Roberts

Despite our recent attempts to resolve your account there is still 320 GBP owed on your account. Your payment terms are 30 Day net.

Please can you contact the credit control department on 01483 8340511 to arrange settlement of this account. If you have recently made a payment then please do accept our apologies.

Yours Sincerely

Imran Sarwar
Credit Collections Dept

26 April 2004 **Page 1 of 1**